

58506

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

VERİTABANLARI ve KÜTÜPHANE OTOMASYONU
ÜZERİNE BİR UYGULAMA

Y. Sâid KENANOĞLU

58506

YAYIN İÇRETİM KURULU
OTOMANTASYON MERKEZİ

Erzurum - 1997

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

VERİTABANLARI ve KÜTÜPHANE OTOMASYONU
ÜZERİNE BİR UYGULAMA

Y. Sacid KENANOĞLU

Yönetici : Prof. Dr. Sibkat KAÇTIOĞLU

Yüksek Lisans Tezi

ÖZET

Bilginin üretim, kullanım ve dağıtımında önemli bir yeri olan kütüphanelerin, gelişen bilgisayar teknolojisinden azami derecede faydalanmaları gerekmektedir. Bu tez ile Atatürk Üniversitesi'nde Merkez Kütüphane Bilgisayar Otomasyon Projesinin gerçekleştirilmesi amaçlanmıştır. Bu proje ile, hem kütüphane servis hizmetlerine hem de okuyucuya yönelik yayın tarama işlemlerine bilgisayar desteği sağlanmaktadır. Program Bilgi İşlem Merkezi'mizde bulunan IBM 4381 mainframe altında geliştirilmiştir. Veritabanı olarak SQL/DS (Structured Query Language/Data System) ve yazılım dili olarak da dördüncü kuşak bir uygulama geliştirme dili olan CSP (Cross System Product) seçilmiştir.



SUMMARY

Of the libraries, which have an important role in the production, use and distribution of the information, they should have to make use from the developing computer technology at the best.

By this thesis, it is aimed in order to realize a project of Central Library Computer Automation at Ataturk University. This project, provides both librarianship services and also computer support for the readers with the aim of searching publication.

Automation program had been developed under IBM 4381 which we have as the mainframe at our Computer Processing Centre. In addition to, SQL/DS (Structured Query Language/Data System) as database and CSP (Cross System Product), which is fourth generation application-development language, as software language have been choosen.



TEŞEKKÜR

Bu çalışmanın ortaya çıkmasında değerli yardımlarını esirgemeyen ve yol gösterici katkılarıyla tezimizin ilmî bir nitelik kazanmasında büyük payı bulunan başta tez danışmanım Sn. Prof. Dr. Sıbkat KAÇTIOĞLU'na, kıymetli hocalarım Sn. Prof. Dr. Fatin SEZGİN'e, Sn. Prof. Dr. Y. Kemal YOĞURTÇU'ya, ayrıca Bilgi İşlem Merkezi'nde görev yapan Uzman Fatih ÖZCAN'a, ekran dizaynlarını bilgisayara aktaran Cahit TOKUYUCU'ya, yol gösterici tutumları ile Doç. Dr. M. Dursun KAYA ve Arş. Gör. M. Macit KENANOĞLU'na ve tezimde emeği geçen tüm arkadaşlara teşekkürlerimi bir borç bilirim.



İÇİNDEKİLER

ÖZET.....	i
SUMMARY.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
1. GİRİŞ	1
1. VERİ (DATA) VE VERİTABANI (DATABASE) KAVRAMI.....	2
1.1. Veri	2
1.2. Veritabanı.....	2
1.2.1. Veri Tanımlama Dili (Data Definition Language-DDL).....	8
1.2.2. Fiziksel, Mantıksal ve Dış Şemalar	9
1.2.3. Mantıksal Dosyalarda Temel İlişki Şekilleri.....	9
1.2.3.1. 1:n İlişki.....	9
1.2.3.2. n:m İlişki	9
1.2.3.3. Bachman Modeli	9
1.2.4. Mantıksal Şemalar	9
1.2.5. Fiziksel Şemalar	10
1.2.6. Dış Şemalar	10
1.2.7. Alt-Şema (Sub-Schema).....	11
1.2.8. Veri İşlem Dili (Data Manipulation Language-DML).....	11
1.2.9. Sistem Destek Fonksiyonları	13
1.2.10. Verinin Korunması (Data Security).....	14
1.2.11. Veri Yapısı (Data Structure).....	14
1.2.11.1. Ardışık Yapılar (Sequential Structures)	14
1.2.11.2. Ağaç Yapılar (Tree Structures).....	14
1.2.11.3. Şebeke Yapıları (Network Structures).....	14
1.2.12. Veri Bağımsızlığı (Data Independence).....	14
1.3. Veritabanı Yönetim Sistemi (Database Management System-DBMS)	15
1.4. Veri Soyutlama.....	16
1.4.1.1. Fiziki Seviye	17
1.4.1.2. Kavramsal Seviye.....	17
1.4.1.3. Görüntü Seviyesi	17
1.5. Veri Modelleri	19
1.5.1. Nesne Tabanlı Mantıksal Modeller (Object-Based Logical Models).....	19
1.5.2. Kayıt Temelli Mantıksal Modeller (Record-Based Logical Models).....	19
1.5.3. Fiziksel Veri Modelleri (Physical Data Models)	19
1.6. Veri Bağımsızlığı	19
1.6.1. Fiziksel Veri Bağımsızlığı.....	20
1.6.2. Mantıksal Veri Bağımsızlığı	20
1.6.2.1. Fiziksel Kat ve Fiziksel Dosyalar.....	21
1.6.2.1.1. Tek Yönlü Bağlı Listeler	22
1.6.2.1.2. Çift Yönlü Bağlı Liste	23
1.6.2.1.3. Ağaç Yapıları	23
1.6.2.1.4. İkiz Çocuk Pointer Yapısı.....	23
1.6.2.2. Mantıksal Kat ve Mantıksal Dosyalar	24
1.6.2.3. Dış Kat ve Dış Dosyalar	24
1.7. Veritabanı Tasarımı.....	24
1.8. Doğruluğun Korunması	24
1.9. Verilerin İşlenmesi	25
1.10. Veritabanı Kullanıcıları	27

1.11. Ayrıntılı Sistem Yapısı	28
1.12. Varlık-İlişki Veri Modeli (Entity-Relationship Data Model)	29
1.13. İlişkisel Model (Relational Model)	32
1.13.1. Görüntüler (views)	36
1.13.2. 4. Normal Form	36
1.14. Şebeke Modeli (Network Model)	37
1.15. Hiyerarşik Model (Hierarchical Model)	39
1.16. Dosya ve Sistem Yapısı	40
1.17. İndeksler	42
1.17.1. Silme Algoritması	44
1.17.2. Araya Ekleme Algoritması	44
1.18. Sorgu İşleme (Query Processing)	45
1.19. Arıza Giderme (Crash Recovery)	47
1.20. Eşzamanlı Kontrol (Concurrency Control)	50
1.21. Dağıtık Veritabanları	51
1.21.1. Dağıtık Sistemin Avantajları	52
1.21.2. Dağıtık Sistemin Dezavantajları	53
1.21.3. İki Safhalı Kabul Protokolü	54
1.22. Güvenlik ve Bütünlüğü Sağlama	55
1.23. Bilgi Tabanları	58
1.24. Çok Ortamlı Veritabanları	59
1.25. Veritabanı Programları	59
1.25.1. İlişkisel Veritabanları	60
1.25.1.1. System R	60
1.25.1.2. Database 2 (DB2)	61
1.25.1.3. Oracle	61
1.25.1.4. Informix	62
1.25.1.5. Progress	62
1.25.2. Şebeke Veritabanları	63
1.25.2.1. Total	63
1.25.2.2. Idms	63
1.25.3. Hiyerarşik Sistemler	63
1.25.3.1. IMS (Information Management System)	63
1.25.3.2. System 2000	64
1.25.4. Mikrobilgisayar Veritabanı Sistemleri	64
1.25.4.1. dBASE-III	64
1.25.5. Spreadsheet Veritabanı Sistemleri	65
1.25.6. Client/Server Veritabanı Sistemleri	65
1.25.6.1. Ingres	65
1.25.7. Windows Veritabanı Sistemleri	67
2. KÜTÜPHANE UYGULAMASI	68
2.1. Kütüphaneler	68
2.2. Kütüphanecilik Hizmetleri	68
2.2.1. Sağlama	68
2.2.2. Hazırlama	68
2.2.3. Yararlandırma	70
2.3. Bilgi, Kütüphane ve Kütüphane Otomasyonunun Önemi	70
2.4. Kütüphane Otomasyonunun Amacı ve Önemi	71
2.5. Kütüphane Otomasyonunun Uygulama Alanları	73
2.6. Kütüphane Otomasyonu Alanındaki Gelişmeler	73
2.7. Türkiye'de Kütüphane Otomasyonu Uygulamaları	74
2.8. Türkiye'de Kütüphane Otomasyonunu Değerlendirme	75
2.9. Uygulama Geliştirmede Kullanılan Araçlar	77
2.9.1. SQL/DS (Structured Query Language / Data System)	77

2.9.1.1. SQL Veritabanının Yapısı	80
2.9.1.2. Mantıksal İş Birimi (Logical Unit of Work-LUW).....	81
2.9.1.3. SQL'nin Otomatik Kilitleme Mekanizması	82
2.9.1.4. İndeksler	82
2.9.1.5. SQL Katalogları	83
2.9.1.6. SQL/DBSU (Database System Utility - Veritabanı Hizmet Programı)	83
2.9.2. CSP (Cross System Product)	84
2.9.2.1. CSP/AD Uygulamalarının Hazırlanması	87
2.9.2.2. CSP/AD'nin Yapısı	88
2.9.2.3. Uygulamaların Tanımlanması	89
2.9.2.4. İşlem İfadeleri	91
2.9.2.5. İşlem Çeşitleri	92
2.9.2.6. MSL Paylaşımı	93
2.9.2.7. ALF (Application Load File)	93
2.9.2.8. Harici Kaynak Biçimli Dosyalar	93
2.9.3. Sistem Analizi	94
2.10. Kütüphane Otomasyonu Uygulama Programı	96
2.10.1. KU00A Uygulama Programı	98
2.10.2. KU01A Uygulama Programı	99
2.10.3. KU02A Uygulama Programı	100
2.10.4. KU03A Uygulama Programı	101
2.10.5. KU04A Uygulama Programı	103
2.10.6. KU05A Uygulama Programı	104
2.10.7. TA10A Uygulama Programı	105
EKLER.....	108
Ek 1. Kullanılan CSP Tabloları.....	109
Ek 2. Kullanılan SQL Tabloları.....	112
Ek 3. Program Yapıları.....	116
Ek 4. Program Ekranları.....	135
Ek 5. Program Akış Şemaları.....	189
Ek 6. Program Yazıcı Çıktıları.....	236
KAYNAKLAR.....	243

1. GİRİŞ

Günümüzde, bilgi çağını yaşadığımız söyleniyor. Bu gelişmenin temelinde özellikle 50'li yıllardan itibaren bilgisayar ve telekomünikasyon alanındaki (iletişim teknolojisinin yaygınlaşması, uydu haberleşmesi, v.b.) başdöndürücü gelişmeler yatmaktadır. Hele hele son yıllarda Internet olarak isimlendirilen global ağ hizmetlerinin artışı ile bilgi iletişiminin doruğuna ulaşılmıştır. Bu sayede dünya küçülmüş, bilgisayar teknolojisinin gelişmesini hızlandırması ile de sosyolojik ve kültürel yapıyı etkilemeye başlamıştır. Teknolojinin bu başdöndürücü gelişmesi ile başlangıçta hayal olarak kabul edilen şeyler hakikat olmuş, yarın ne olacağı ise tahmin sınırlarının ötesinde kalmıştır.

Bilgisayar teknolojisindeki bu gelişme hem bilgi üretimini artırmış hem de üretilen bu bilginin iletişimde büyük kolaylıklar sağlamıştır. Tüm bunlara paralel olarak da yepyeni bir bilgi dalı gelişmektedir. Bilgi Bilimi (Information Science). Bu bilgi dalı bilgi merkezlerinde toplanacak bilgileri ihtiyaç duyulduğu anda ilgililere en kısa zamanda ulaştırmanın yollarını araştırmaktadır. Bugün Pathfinder'in Mars'tan gönderdiği bilgilere sadece ilk günlerde bile 100 milyondan fazla kişinin ulaştığı bildirilmektedir.

Mektup, telgraf, teleks gibi iletişim araçları artık mazide kaldı. Şimdi tele-işlem, tele-konferans, tele-bankacılık, tele-ansiklopedi, tele-pazarlama, tele-tahmin, v.b. devri. Tüm bu gelişmeler toplumsal hayat tarzını ve alışkanlıkları etkileyip değiştirmektedir.

Bilgi kapalı ve sınırlı kaldığı sürece faydalı olmamaktadır. Faydalı olması için paylaşıp ilgili kişiler ve kurumlar arasında transfer edilmesi gerekir. Araştırmacılardan, öğrencilerden, bilim adamlarından, analiz yapanlardan tutun ticari şirketlere kadar herkes ihtiyaç duydukları bilgiyi en kısa sürede ve en doğru bir biçimde elde edebilmek için büyük gayretler sarfetmektedirler. Çünkü artık bilgi kaynakları sınırlı değildir. "Aranılan bilgi nerededir? Ona nasıl ulaşılır?" sorularının cevaplandırılmasında bilgisayar teknolojisi vazgeçilemez bir hususiyet kazanmıştır. Konunun uzmanları bilgiye erişim ve transfer etme probleminin, merkezi veri bankalarının kurulması suretiyle zaman ve araştırma maliyetini azaltıp, sistem yaklaşımına dayanan modern bir yönetim anlayışı ile çözülebileceğini söylemektedirler.

Bu veritabanı veya veri bankalarının en önemlilerinden birisi de kütüphanelerdir. Bilginin üretim, kullanım ve dağıtımında otomasyon ve kütüphanecilik birbirine paralel ve birbirini destekleyerek gelişmek zorundadır. Bu entegrasyon tam olarak sağlandığında oluşan bilgi kirliliğinin de sona ereceği kanaatindeyiz. Nitekim ODTÜ'de yapılan ve şahsen iştirak ettiğimiz bir kongrede de bu tür bir çalışma içine girildiği görülmüştür.

Çalışmamız iki bölümden meydana gelmektedir. İlk bölümde veritabanı konusu teorik çerçevede ele alınmış olup, ikinci bölümde kütüphanecilik servisleri ve kütüphane otomasyon programının araç ve aşamaları yer almaktadır. Ekler bölümünde bu programa ait tablolar, yapılar, akış şemaları ve çıktı örnekleri bulunmaktadır.

1. VERİ (DATA) VE VERİTABANI (DATABASE) KAVRAMI

1.1. Veri

“Kimi yerde data, kimi yerde veri olarak karşılaştığımız bu kelimenin farklı tanımları mevcuttur. Webster's II sözlüğüne göre data "Bilgisayarda işlenmek üzere toplanmış sayısal enformasyon, datum kelimesinin çoğul hali, analiz için derlenmiş enformasyon" olarak tanımlanmaktadır. D.S. ve K.M. Hussain, Information Processing Systems for Business adlı kitaplarında "Data durum veya gözlemlerin gösterimidir" demektedirler. Bu tanımları biraz daha açıp datayı yeniden tanımlayacak olursak ;

Data; Analize temel teşkil edebilecek herhangi bir anın, faaliyetin, durumun, nesnenin ya da bütün bunların tamamının sayı, harf ve simgeler grubunun bir veya daha fazlası kullanılarak önceden kabul görmüş kurallar çerçevesinde yapılan tanımının herhangi bir ortamda kayıtlı bulunması halidir. Bilginin hammaddesi olarak da tanımlanabilmektedir.

Latince kökenli kelimenin tekil hali datum olup, günümüzde hem tekil hem de çoğul hali için kelimenin çoğul şekli olan data kullanılmaktadır. Data kendi içerisinde bir ayrıma - düzenli ve düzensiz olarak - tabi tutulabilir. Belirli bir tarama işleminden geçirilmemiş data işlenmemiş/düzensiz (raw) data olarak adlandırılır. Sıralama, ayıklama v.b. işlemlerden geçirildikten sonraki yeni biçimine işlenmiş/düzenli (processed) data adı verilmektedir. Bu tür data değişik ortamlarda saklanabilir, kullanılabilir.

Sümerlerin kil tabletelerinden, zaman içinde değişik ortamlara geçildiyse de (papirüs, kağıt, delikli kartlar, manyetik ortamlar v.b.), veri her zaman o günkü değerlerden enformasyon (information) ve bilgi (knowledge) çıkarmamıza kaynak teşkil etmiştir. Anket yapan kişinin sorularına aldığı cevapların tümü veridir. Bilgisayara belli bir düzende aktarıldığında da veridir ve veri kimliğinden ancak anlamlı sonuçlar çıkarıldığı zaman bir enformasyon haline geçmekte, kişiler tarafından özümsemiğinde ise bilgi niteliğine bürünmektedir. Bilgi niteliğine kavuşan veri aslında özümsemiği ortama göre farklı biçimler aldığından bilgi tanımı kimi yazarlar ve araştırmacılar tarafından birikim olarak adlandırılmaktadır¹.”

Veriler, sayısal veya alfasayısal (karakter) türde olmak üzere iki grupta ele alınmaktadır.

1.2. Veritabanı

70'li ve 80'li yılların başlangıcında daha yeni ve anlaşılabilir olan veritabanları ilerleyen yıllarda hayatımıza daha sık girmeye başladılar ve günümüzdeki elektronik veri yönetiminin merkez kavramı haline geldiler. Finans, depo yönetimi, pazarlama bölümü, bir işletmedeki diğer işler olsun veya kamudan özel sektöre başka konular olsun veritabanları vazgeçilmez yardımcılar haline geldiler².

Veritabanı kavramı değişik kaynaklarda şu şekilde tarif edilmektedir.

¹ Zafer BABÜR, 1995, Bilgisayarla İletişim, Sistem Yayıncılık, İstanbul, ISBN:975-7397-50-4, s. 1-3

² Anonymous, Nisan 1996, CHIP Bilgisayar Kültürü Dergisi, Yazılım İnceleme (Veritabanı), s. 72

“Veritabanı, özel bir amaca hizmet etmek üzere düzenlenen ve sunulan bir bilgi birikimi veya belirli bir tarzda organize edilmiş faydalı bilgi koleksiyonudur³.”

“Veritabanı, bir konu veya kuruluşa ait verilerden oluşan ve genellikle Veritabanı Yönetim Sistemi (Database Management System-DBMS) adı verilen gelişmiş yazılımlar ile yönetilen veri kümeleri topluluğudur⁴.”

“Veritabanı, birçok kullanıcı tarafından kullanılan birbirleriyle ilişkili geniş bir veri kümesinin düzenlenmesi, depolanması ve sorgulanması için kurulan bir sistemdir⁵.”

“Kullanımları bir şema ile tanımlanmış olan set, kayıt ve alanlardan oluşan sistem⁶.”

Veritabanlarına örnek olarak telefon rehberi, sözlük, yemek kitabı, posta sipariş katalogları, ansiklopedi, kütüphane katalog kartları, gazetelerde yer alan döviz ve borsa haberleri, şirket bilançoları verilebilir. Telefon dizininde, telefon numaraları ve adresler isimlerle ilgili tutulmaktadır. İsimler alfabetik sıra içerisinde verildiğinden bunları kolaylıkla bulabilirsiniz. İsmi bulunca adres ve telefon numarasına da erişirsiniz. İsim telefon rehberini kullanmak için bir anahtar olmaktadır. Sözlüklerde ise bir kelime ve bir tanım bulunmaktadır. Kelimeler kolaylıkla bulunabilmeleri için alfabetik olarak sıralanmıştır. Tanımlar ise o kelimeyle ilgili olarak verilmiştir. Dolayısıyla sözlüğün kullanım anahtarı kelime olmaktadır. Verilen örneklerin tümünde yer alan ortak nokta düzenlenen bilginin bir anahtar yoluyla kolaylıkla bulunabilmesidir. Başka bir deyişle bir cetvel halinde sunulan (sıralar ve sütunlar olarak) herhangi bir bilgi, bir veritabanı olabilir⁷.

Bilgi sistemlerinin uygulanmaya başlaması, basit kayıt işleme sistemlerinin uygulamaya konmasıyla başlamıştır. Örgüt içindeki değişik birimlerin, kendi ihtiyaçlarını karşılayacak kayıt işlem sistemlerini kurmaları, birbirinden bağımsız olarak bu sistemleri bilgisayar ortamına aktarmaları ve geliştirmeleri sonucu etkin olmayan sistemler ortaya çıkmıştır. Bu yaklaşımın en önemli eksikliği niteliklerin, değerlerin ve gösterimlerin standartlaştırılmış olmamasından kaynaklanmaktadır. Dolayısıyla, hem örgüt içinde biriken bilgilerin değişik örgüt birimleri tarafından kullanılabilme ihtimali düşmekte hem de veriye ulaşabilmek için fazladan arama ve sıralama işlemleri yapmak gerekmektedir.

Uygulama yaklaşımının kullanıldığı veritabanı sistemlerinde ortak veri elemanları mantıksal olarak birbirlerine bağlanmamışlardır. Fakat sistem yaklaşımının kullanılmasıyla oluşturulan veritabanı sistemlerinde ise ortak veri elemanlarının mantıksal olarak birbirlerine bağlanması esastır. Böylece hem güncelleştirmeler eşzamanlı yapılabilmekte hem de sistemin aynı veri için değişik değerler içermesi önlenmiş olmaktadır. Öte yandan

³ Anonymous, 1989, DBASE IV, Ekonomist Yayınevi, Ankara, s.7

⁴ Anonymous, 1991, Bilgisayar Ansiklopedisi, Milliyet Yayınları, s. 192

⁵ Anonymous, CHIP, a.g.d., Nisan 1996, sh. 72

⁶ Emin D. Aydın, 1990, Veritabanı (Database), İstanbul, Evrim Basım Yayım Dağıtım, s. 56

⁷ Anonymous, DBASE IV, a.g.e., s.7

sistemde oluşan kayıt tekrarları de yine bu yaklaşım aracılığı ile asgari seviyede tutulmaktadır⁸.

Geçmişten günümüze daha merkezî ve denetimli veri kontrolü sağlama adına, veri üzerinde doğrudan kontrol mekanizmalarının programcılar tarafından terk edildiğini görüyoruz. Bu gelişme, yalnız bir işlem birimi tarafından veri erişiminin yapıldığı klâsik bilgi işlem felsefesinde de paralel bir değişime neden olmuştur. Bu değişimle birlikte, işlem biriminden tam performans almak ve veri üzerinde diğer işlemlerin de yapılabilmesini sağlamak için veri kütüklerinin tekrar sıralanıp, yine aynı verilere sahip gereksiz ve yeni kütüklerin açılması yoluna gidilmiş, işlem kökenli kullanım tekniği geleneksel bir şekil almıştır. Bu teknik bazı durumlarda yeterli olmakla beraber, genellikle de pahalı ve pratik olmayan bir çözümdür. Günümüzün bilgi depolama birimlerinin, aynı verinin birden çok uygulama programı tarafından performans kaybına yol açmadan paylaşılabilirdiği entegre bir yaklaşım gerektirmesi, pahalılığın başlıca sebebidir. Fiyat faktörünün bu kadar önemli olmasının sebebi de genellikle idare ve korunmalarının çok zor olması gösterilmektedir. Verinin birden çok işlem birimi ve uygulama programı tarafından kullanılmasını ve dolayısıyla işlem birimlerinin birbirlerinden haberdar olmasını gerektiren ortamlarda klâsik teknik, gittikçe pratik olmaktan uzaklaşmaktadır⁹.

Geleneksel veri kütükleri ile çalışıldığında, uygulama programcısı verinin ne şekilde örgütlendiğini, veri kayıtlarının biçimini, kütük hacmini, kayıt uzunluğu ve kütüğün fiziksel özelliklerini bütünüyle bilmek zorundadır. Veri kütüğünün özellikleri herhangi bir sebepten ötürü değiştirildiğinde -kütüğün fiziksel özellikleri değiştiği için- bu kütüğü kullanan uygulama programlarının da uygun şekilde değiştirilmesi gerekmektedir.

Oysa veritabanı yaklaşımında kullanıcılar yalnızca programlarda gerekli olan verilerin neler olduğunu bilmekle yükümlüdürler. Veritabanının fiziksel karakteristikleri ya da hangi bellek adresinde olduğunu bilmesi gerekmemekte, veritabanında yapılan bir çok düzenleme ve değişiklik kullanıcıyı ilgilendirmemektedir.

Bilgi toplumuna geçiş süreci içerisinde üretilen yığın bilgileri tekrarsız biriktirebilmek ve tüm kullanıcılara açık duruma getirebilmek, temel ihtiyaçlar olarak ortaya çıkmaktadır. Bu temel sebeplerden ötürü veritabanı (database) kavramı gündeme gelmektedir. Hazırlanacak olan veritabanı, bilgilerin nasıl kullanılacağı önceden bilinemediği için dinamik bir bilgi tanım ve erişim düzeni getirmeli, bilgilerin anında oluşturulup değiştirilebildiği ve ilişkilendirildiği çevrim içi (on-line) çalışma imkânı vermelidir. Aslında veritabanı, verilerin örgütte çok sayıda uygulama tarafından kullanılabilmesine imkân verecek şekilde örgütlenmesi ve denetlenmesidir. Farklı uygulamaların aynı verilerden yararlanabilmelerini, gereksiz ve tekrarlı verilerin ayıklanmalarını sağlamakta, bu işlemleri yaparken veri ile, verinin saklı olduğu ortam arasındaki bağımlılığı en aza indirmektedir¹⁰.

⁸ E. Akmen ve I. Nazlıarslan, Kasım 1989, SQL/DS, Anadolu Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi, s. 2

⁹ Emin D. Aydın , a.g.e., s. 9

¹⁰ Akmen ve Nazlıarslan, a.g.e., s. 1

Veritabanlarının önemi ve faydası genellikle çok kullanıcıli bilgisayarlarda daha da açık olarak ortaya çıkmaktadır. Bu tür sistemlerde bir dosya bir çok kiři tarafından aynı anda kullanılmakta yani bilgiler paylaşılmaktadır. Böylece herkes için aynı dosyaları tekrar tekrar oluşturmak ve sürekli güncellemek gerekmemektedir. Bu oldukça önemli bir özelliktir. Bilgilerin paylaşılmasını, İşletim Sistemi ve Veritabanı Yönetim Sistemleri ortaklaşa başarmaktadırlar.

Veritabanındaki bilgilerin paylaşılabilmesi de önemli bir özelliktir. Böylece bilgiler üzerinde merkezî bir kontrol sağlanabilmekte ve standartlar korunabilmektedir. Veritabanındaki dosyaların içeriklerinin kullanıldığı sürece mümkün olduğu kadar doğru olarak saklanması çok önemlidir. Bunun için pek çok önlem alınabilir. Bunların başında dosyaların doğru olarak tasarlanması gelmektedir. Dosya tasarımları için farklı modeller geliştirilmiş, geliştirilen yeni modeller bir öncekinin üzerine yeni özellikler eklemiş fakat ana modelin çok önemli özelliklerini bozmamıştır¹¹.

Bir örgütün bütün verilerinin entegre bir veritabanında saklanması için pek çok gerekçe sayılabilir. Bütün bir veritabanının örgüt yöneticilerine sağlayacağı merkezî denetim yeteneğinin avantajları şu şekilde özetlenebilir¹².

1- Gereksiz veri saklama azaltılabilmektedir. Entegre olmayan bir sistemde aynı verinin pek çok ayrı birimde saklanması ve güncellenmesi gerekebilir. Bu durumda hem fazladan bir maliyet söz konusu olmakta, hem de -verinin farklı güncellenmesi sebebiyle- aynı veri için farklı değerler oluşmaktadır.

2- Saklanan verinin paylaşılması mümkün olmaktadır. Örgüt birimlerinin herhangi birine gerekli olduğu için saklanan bir veri, başka bir örgüt birimine de gerekebilmektedir. Böylece entegre olmayan bir sistemde sık sık başvurulmadığı için saklanması ekonomik olmayan bir veri, entegre bir ortamda bu bilgiye sık sık ihtiyaç duymayan birimler için ulaşılabilir olmakta ve birimlerin etkinliği artmaktadır.

3-Standartların konulması sağlanmaktadır. Entegre bir sistemde veritabanındaki verilerin yaygın standartlara uymasını sağlamak kolaylaşır. Bu durumda verinin elde edilmesi bilgisayar ortamına aktarılması, bakımı, güncellenmesi ve değişik örgüt birimleri tarafından paylaşılması kolaylaşır.

4-Gizlilik sınırlamalarının uygulanması kolaylaşmaktadır. Sistem bir bütün olduğu ve merkezî bir şekilde denetlendiği için veriye erişim ve güncelleme konusunda değişik standartların tesbiti ve uygulanması daha kolay bir hale gelmektedir.

¹¹ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 193

¹² Akmen ve Nazlıarslan, a.g.e., s. 2

5- Çelişen tercihler arasında daha uygun bir denge sağlanabilmektedir : Örgüt içindeki birimler, bilişim sisteminin kendi tercihleri doğrultusunda hizmet vermesini isterler. Merkezî denetim sayesinde çelişen bu tercihler arasında örgütün tamamı lehine daha uygun bir dengenin tesbit edilmesi ve uygulanması mümkün olmaktadır.

Veritabanı sistemlerinin en önemli bir avantajlarından biri de veri bağımsızlığını sağlamasıdır. Entegre olmayan sistemlerde veri yapısı, uygulama programlarının ihtiyaçları göz önüne alınarak düzenlenir. Bu durumda, verinin değişik bir görünümüne gerek duyan yeni uygulama programlarının düzenlenmesi imkânsızlaşır. Öte yandan uygulama programlarını değiştirmeden verinin yapısını değiştirmek, entegre olmayan sistemlerde mümkün değildir¹³. Sistem tasarımlarının istenen esnekliğe, genişliğe sahip olmasını gerektiren bir veritabanı sistemi ise aşağıdaki özellikleri içermelidir.

- 1- Bütün uygulama programlarına göre bir veri yapısı standardı olmalıdır. Veri aynı anda bir çok programa uygunluk özelliğine sahip olmalı ve bunu gerek programları, gerekse veritabanını gereksiz çevrimlere sokmadan yapmalıdır.
- 2- Verinin birden çok işlem birimi tarafından güncelleştirilmesini veya okunmasını sağlamalıdır.
- 3- Veritabanının tümü veya bir bölümü için çeşitli veri arama stratejilerine sahip olmalıdır.
- 4- Veritabanının yetkisiz kullanımına ve yanlış erişim taleplerine karşı güvenlik sistemi olmalıdır.
- 5- Veritabanındaki mevcut verinin fiziksel yerleşimi üzerinde kontrol sağlamalıdır.
- 6- Programlar için aygıt bağımsızlığı sağlamalıdır.
- 7- En küçük veri parçasından, geniş bilgisayar şebekelerine kadar bir yelpaze içinde veri yapılarının geçerliliğini sağlamalıdır.
- 8- Geçerli tekniklerin elverdiği ölçüde programların veriden bağımsızlığını sağlamalıdır.
- 9- Veritabanı için belli bir program dilinden bağımsız tanımlama bölümü olmalıdır.
- 10- Birden çok programlama dilini kullanan ve veritabanını tanımlamaya imkan veren bir mimariye sahip bulunmalıdır.

Bu özellikler uygulama programları için veritabanı esnekliğini ve veritabanının genel kullanıma açıklığını sağlamakta ve ancak uygulama programlarının izin verdiği kadar karmaşık veri yapılarının kullanımına imkân vermektedir. Gerçi bu veri yapıları ile bir tekniğin tek başına yeterli problem çözücü modelleri sağlaması zordur. Ancak problemin bütünü için bir kaç teknik bir bütün oluşturacak şekilde kullanılabilir. Bu bütünlüğü veri yapısı sağlamaktadır. Seçimi yapılacak veri işlem dilleri geliştirilmiş diller olabilir. Günümüzün bu tip "kendi başına" dil sistemlerinin başlıca yetenekleri şunlardır.

- 1-Sorgulama : Veri seçimi, sıralama ve raporların biçimlendirilmesini yapmak.
- 2-Güncelleştirme : Güncelleştirilecek verinin seçimi ve seçilen verinin değerini değiştirme işlemi yapmak.
- 3-Oluşturma : Veritabanının tamamını veya bir kısmını oluşturma işlemlerini yapmak.

¹³ Akmen ve Nazlıarslan, a.g.e., s. 2

4-Yeniden yapılandırma : Halen var olan veritabanı tanımını deęiřtirme ve veritabanını yeni tanıma göre hazırlama iřlemlerini yapmak.

Programlama esnasındaki zaman ve emek tasarrufu bu dillerin göz ardı edilmeyecek avantajlarıdır. En önemli dezavantajları ise uygulama alanlarının darlığı yüzünden sınırlarının, fonksiyonel yeteneklerinin ve esnekliklerinin çizilmiş olmasıdır. Kuřkusuz bu performans farkları, iřlenecek veri miktarının çok olduęu, iřlem sayısının arttığı veya iřlemin tekrar edildięi uygulamalarda önem kazanmaktadır. Sonuç olarak hem programcı olanları hem de olmayanları veritabanı ile birleřtirebilecek bir tasarım yaklařımı gerekmektedir. Kullanıřlı bir veritabanı oluřturmayı amaçlayan her kuruluřun katetmesi gereken er veya geç uzun bir yol vardır. Yapılması gereken, belirlenmiř amaçlara yönelik bireysel sistemleri gerçekteřtirmek, yöntem sistemlerini bilgi sistemlerine tercih etmek ve somut sonuç verebilecek olanları seçmek olacaktır¹⁴.

Veritabanı bir çok uygulamalara servis hizmeti verdięi için veritabanı yöneticisi tarafından yürütölen merkezî bir koordinasyona ihtiyaç vardır. Bu tek bir kiři olabileceęi gibi her bir veritabanından sorumlu yöneticiler de olabilmektedir. Veri yöneticisi řirketin veri kaynakları için yapılan genel stratejiden sorumlu olup řirket veritabanı sistemlerinin geliřmesini kontrol eden kiřidir. Veritabanı yöneticisinin ilgi alanı ise bir servis fonksiyonundan kuruluřun tüm verilerini saklamaya kadar uzanabilir. Veritabanı yöneticileri belirli bir veritabanına ait iřlemlerin yürütölmesi gibi teknik bir iře sahipken, veri yöneticisi strateji ve planlamayla ilgilenir¹⁵.

Bilgisayar veritabanı kütöüklerini daha iyi tanımak için bilgilerin planlanması, bilgilerin oluřturulması ve bilgilerin deęiřtirilmesi gereklidir. Bu iřler için bir yönetici sistem - Veritabanı Yönetim Sistemi (Database Management System-DBMS)¹⁶- gereklidir. DBMS'ler veritabanı kütöüğündeki tüm ayrıntıları dikkate alan, çok büyük sayıda veri elemanlarını etkili bir řekilde idare eden güçlü bilgisayar programlarıdır ve bir kütöüğün içeriğini kullanmayı, yönetmeyi ve deęiřtirmeyi saęlamaktadır. Günümüzde devamlı olarak deęiřen bilgi alıř-veriři en basit sistemler için bile ilgili bir veritabanı yöneticisi gerektirmektedir. DBMS bir bilgisayar içine yerleřtirildiğinde o bilgisayar depolama, kataloglara ayırma ve veriyi yeniden düzenleme gibi ayrıntılar üzerinde beceri kazanmaktadır. Sizin yapacaęınız řey ne istedięinizi bilmektir. İřin geri kalanını DBMS yürütecektir. Bu programlar gerçekten bir araçlar seti olup - basit olmakla beraber - etkili bir řekilde veriyi organize edebilir ve iřleyebilirler. DBMS uzun zaman büyük bilgisayar sistemleri (mainframe) içinde büyük bilgi koleksiyonlarının sistematik organizasyonu ve idaresinde kullanılmış ise de günümüzde bu terim daha genel olarak bilinmeye bařlamıřtır. Sonuçta kiřisel ve iř uygulamaları için daha fazla DBMS yazılımları geliřtirilmiřtir¹⁷.

¹⁴ Emin D. Aydın , a.g.e., s. 10-11

¹⁵ Emin D. Aydın , a.g.e., s. 37

¹⁶ İlerleyen bölümlerde Veritabanı Yönetim Sistemi "DBMS" olarak anılacaktır.

¹⁷ Anonymous, DBASE IV, a.g.e., s. 10-11

Kısaca özetleyecek olursak bir veritabanı iki parçadan, veritabanı ve DBMS'den oluşmaktadır. Veritabanı kaydedilen verileri tutarken, DBMS veri giriş, yönetim ve değerlendirme operasyonlarını icra etmektedir. DBMS'ler verileri yapılarına göre bir veya daha fazla dosyada saklamaktadırlar. Örneğin bir DBMS yazılımı olan dBase her tabloyu ayrı bir dosyada ".DBF" uzantısıyla, Access ise tüm verileri ".MDP" uzantılı tek bir dosyada saklamaktadır¹⁸. Bir veritabanı içinde depolanan bilgiler organize edilerek birkaç şekilde görülebilmektedir. Bilgi işlem ortamında genel olarak karşımıza üç tip veritabanı modeli -Hiyerarşik (Hierarchical), Şebeke (Network), İlişkisel (Relational)- çıkmaktadır¹⁹.

Veritabanılarına ait detaylara inmeden önce tezin ilerleyen bölümlerinde karşılaşılabilecek olan kavramlardan bazılarını kısaca değineceğiz.

1.2.1. Veri Tanımlama Dili (Data Definition Language-DDL)

Bir veritabanı şeması, veri tanımlama dili olarak isimlendirilen ve özel bir dil olarak açıklanan tanımlar topluluğunca belirtilir. DDL ifadelerinin derlenmesi sonucu oluşan tablo seti, özel bir dosya olan veri sözlüğünde (data dictionary²⁰ veya directory) saklanır. Bu tablo, veri hakkında veri (metadata) içeren bir dosyadır. Veritabanındaki gerçek bir veri okunmadan veya değiştirilmeden önce bu dosya tahlil edilir. Veritabanı sisteminin kullandığı erişim metotları ve depolama alanı, DDL'in özel bir şekli olan bu veri depolama ve tanımlama dili tarafından açıklanır. Bu tanımların derlenmesi ile, genellikle kullanıcılardan saklanmış olan veritabanı şemalarının gerekli detaylarını açıklayan bir komut seti elde edilir.

Veri birimi (data item) : Verinin en küçük parçası olup, veri biriminin varlığı bir değer var olması anlamına gelmektedir.

Veri öbeği (data block) : Bir birim olarak düşünülüp işlem gören karakter, bayt, veri topluluklarıdır. Fiziksel kayıt olarak da adlandırılır.

Kayıt (record) : Veri birimlerinin ve öbeklerinin topluluğundan oluşmaktadır. Kayıt ve kayıt tipi farklı farklı kavramlardır.

Set : İsimlendirilmiş kayıt topluluklarına verilen isimdir.

Alan (field) : Veritabanında adreslenebilir kayıt saklama bölgesinin bir alt bölümüdür.

Şema (schema) : Şemalar, veritabanı dili girişlerinden oluşmaktadır ve bir veritabanı için tam bir tanımlayıcıdır. Şema içerisindeki bütün alan, set oluşumları ve veri öbeklerinin veritabanı içindeki durumları ile isim ve tanımları bulunmaktadır²¹.

Veri Sözlüğü (data dictionary) : Bir DBMS içerisindeki tüm kütük, alan ve verilerin listelendiği tabloya verilen isimdir.

¹⁸ Anonymous, CHIP Bilgisayar Kültürü Dergisi, a.g.d., s. 72

¹⁹ Bu modeller ileride teferruatlı olarak ele alınacaktır.

²⁰ Chris Gane and Trish Sarson, 1979, Structured System Analyses : Tools and Techniques, Prentice-Hall, Inc., Englewood Cliffs, New Jersey U.S.A., ISBN : 0-13-854547-2, Chapter 4, p. 48-75

²¹ Emin D. Aydın , a.g.e., sh. 55

1.2.2. Fiziksel, Mantıksal ve Dış Şemalar

Şemalarda, fiziksel, mantıksal ve dış seviyedeki dosyaların özellikleri belirlenmektedir. Bu şemaları oluşturmak için şema dilleri kullanılmaktadır. Yazılan kaynak şemalar derlenince hedef şemalar oluşmaktadır. Bazı dillerde ise etkileşimli programlar vasıtasıyla şemalar tanımlanmaktadır.

1.2.3. Mantıksal Dosyalarda Temel İlişki Şekilleri

Dosyalar arasında iki farklı temel ilişki şekli mevcuttur. Bunlar 1:n ve n:m ilişkilerdir.

1.2.3.1. 1:n İlişki

Dosyadaki bir elemanın (satır veya kayıt) diğer bir dosyada birden çok eleman ile ilgili olduğunu göstermektedir. Örnek vermek gerekirse bir dosyada pek çok deponun kaydının olduğunu, diğer bir dosyada ise o kuruluştaki çalışan personele ait bilgilerin tutulduğunu düşünelim. Bir depoda birden çok personel bulunduğundan depo dosyası ile personel dosyası arasında 1:n bir ilişki mevcuttur. Bazı depolarda bir personel olması, bazılarında hiç bulunmaması 1:n ilişki özelliğini bozmamaktadır. Bir depoda birden çok personel vardır ama bir personel için tek bir depo söz konusudur.

1.2.3.2. n:m İlişki

Bu tür ilişkili dosyalardaki bir eleman ikinci dosyadaki birden çok eleman ile, ikinci dosyadaki bir eleman da birinci dosyadaki birden çok eleman ile ilişkilidir. Örnek olarak birinci dosya için personel dosyasını, ikinci dosya için personelin depolar için sipariş verebileceği satıcıların dosyasını düşünelim. Bir personel, satıcı dosyasındaki birden çok satıcıya siparişini vermiş, aynı şekilde bir satıcıya birden çok personel tarafından sipariş gelmiş de olabilir. Bu durumda satıcı dosyası ile personel dosyası arasında n:m bir ilişkiden söz edilmektedir.

1.2.3.3. Bachman Modeli

Sadece mantıksal dosyaların bazı özelliklerinin ve aralarındaki ilişkilerin gösterildiği bir grafik modeldir. Bachman modelinde sadece 1:n ilişkiler gösterilmekte, n:m türü ilişkiler ise iki adet 1:n ilişki oluşturularak gösterilmektedir. Farklı mantıksal dosyalar arasındaki ilişki 1:n veya n:m olabilir. Fakat dosyaların tasarımı yapılırken n:m ilişkiler, 1:n ilişkilere indirgenmiş olarak kullanılır.

1.2.4. Mantıksal Şemalar

Mantıksal dosyalar farklı dillerde farklı şekillerde tanımlanmaktadır. Örneğin SQL dili ile yazılmış bir mantıksal dosya tanımı aşağıdaki gibidir²².

```
Create table depo
  (Depo_no char (5) not null,
   Sehir char (12),
   Hacim decimal (8) )
Primary key depo_no in insaat
```

²² Anonymous, Bilgisayar Ansiklopedisi, a.g.e.. s. 196-202

Aynı dosyanın IMS veritabanı yönetim sistemindeki tanım biçimi ise aşağıdaki gibidir.

```
01 Dbd name=insaat
02 Ssegm name=depo,bytes=25
03 Field name=(depo_no, seq), bytes=5, start=1
04 Field name=sehir,bytes=12,start=6
05 Field name=hacim,bytes=8,start=18
```

CODASYL yaklaşımlı DBMS'lerde ise tanım aşağıdaki gibi yapılmaktadır²³.

```
01 Schema name is insaat
02
03 Record name is depo
04 Key depo_no is ascending depo_no
05 Duplicates are not allowed
06 02 Depo_no pic x(5)
07 03 Sehir pic x(12)
08 04 Hacim pic 9(8)
```

1.2.5. Fiziksel Şemalar

Dosyalara erişim metotları, fiziksel şemalar tanımlanarak belirlenmektedir. Örneğin SQL'de bir indeks tanımı aşağıdaki gibidir. Bu ifade depo dosyası için DEPONDX isimli indeks dosyasını depo dosyasındaki DEPO_NO alanı üzerinden yapmaktadır.

Create index depondx on depo (depo_no)

Aynı amaç için dBASE-IV dilinde ise “**index on depo_no to depondx.ndx**” komutu kullanılmaktadır. Birkaç alanı birden içeren indeksler oluşturulabilir. Satınalma dosyası için oluşturulacak bir indeksin SQL dilindeki karşılığı ;

Create index satndx on satinalma (personel_no, malzeme_no)

1.2.6. Dış Şemalar

Dış şemalarda ise dış dosyalar tanımlanmaktadır. SQL dilinde bir dış şema tanımı aşağıdaki gibi yapılmaktadır.

```
define view view1
    (depo_no, personel_no, sehir, isim, maas, hacim)
as select depo_no, personel_no, isim, maas, sehir, hacim
from depo, personel
where personel.depo_no = depo.depo_no
```

veya

²³ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 203-205

```
define view view2
    (depo_no, personel_no, isim)
as select depo_no, personel, isim from depo
```

1.2.7. Alt-Şema (Sub-Schema)

Alt-şemalarda veritabanı girişleri bulunmakla beraber, veritabanının bütünü için bilgi bulunmaz. Sadece belli programların kullandığı set, kayıt, alan, veri birimi ve öbekleri hakkındaki bilgileri içermektedirler. Alt-şema, alt kümesi olduğu ana şemanın mantıklı ve devamlı bir elemanı olmak durumundadır. Bir uygulama programı alt-şema ile sınırlı durumdadır. Daha geniş düşünülecek olursa bu sınırlılık, programın veritabanının geri kalan bölümünden yalıtımını ve bütünlüğünü, kısmen de veri bağımsızlığı sağlayabilmektedir. Veritabanı şemasında, dolayısıyla verilerde bazı değişiklikler yapılabilmesi o veriyi kullanan programın kullanmakta olduğu alt-şemanın, asıl şemadan farklılık gösterebilme özelliği ile mümkün olmaktadır. Şema ve alt-şemaların bir takım özellikleri bulunmaktadır:

- 1- Bir şema kaynak kodu herhangi bir uygulama programına veya alt-şemaya bağlı olmaksızın derlenebilir.
- 2- Bir alt-şema kaynak kodu ise herhangi bir uygulama programından bağımsız olarak derlenip saklanabilir.
- 3- Bir şema bünyesinde birkaç alt-şema deklarasyonu olabilir.
- 4- Her alt-şema isimlendirilmelidir.
- 5- Bir alt-şemayı uygulama programı işletir.
- 6- Aynı alt-şema değişik programlar tarafından uyarılabilir.
- 7- Her program bir alt-şema içinde kendine ait alan, kayıt, veri birim ve setlerini kullanabilir.
- 8- Alt şemalar servis diline dayalı olarak oluşturuldukları için bir program kendi kaynak diline uygun olan alt-şemayı uyarmalıdır.

1.2.8. Veri İşlem Dili (Data Manipulation Language-DML)

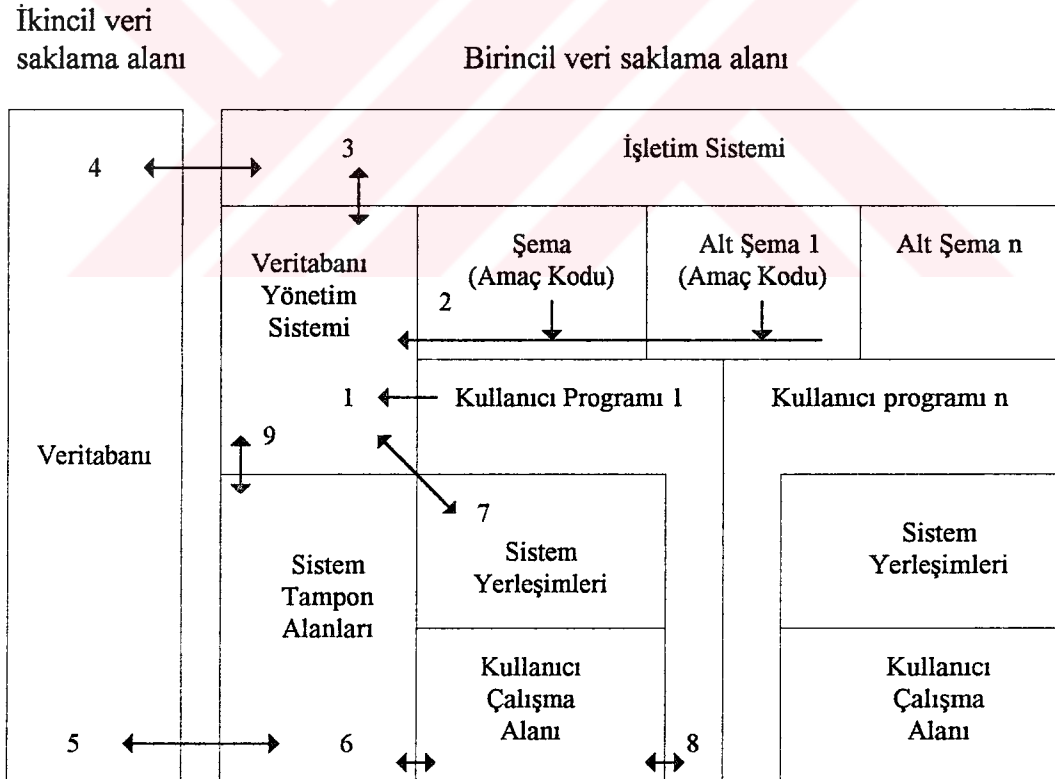
Kullanıcı, program ve veritabanı arasında veri transferi sağlayan ortama veri işlem dili denilmektedir. Bu dil başlı başına bir programlama dili olmayıp bir servis diline ihtiyaç göstermektedir. Yetenekleri de bu servis dilinin yordamları ile sınırlıdır. Bir kullanıcı uygulama programı; veri işlem dili ve servis dilinin karışımı ile yazılmaktadır. Veri işlem dili veritabanı ile iletişim sağlamakta, başka bir deyişle DBMS'in çalıştığı dil olmaktadır. Bir veriyi okumak, silmek, güncelleştirmek veya yeni verileri eklemek için veritabanına/veritabanından yapılan çağrılar ve bunlar arasındaki ilişkiler yine bu dilde yazılmaktadır. Böylece kullanıcı program yazmaya gerek duymadan da verileri kullanabilmektedir²⁴.

Başarılı bir çağrı işletiminden sonra veri, çağrıyı yapan programın kullanıcı çalışma alanına aktarılmakta ve burada servis dilinin gereklerine göre işlem görmektedir. Yeni bir veri eklemek veya değiştirilmiş veriyi tekrar veritabanına göndermek, kullanıcı çalışma alanı içinde bazı parametrelere gerekli değerleri vermek amacıyla servis dili, bu transferle

²⁴ Emin D. Aydın , a.g.e., s. 56

ilgili DBMS'den servis çağrılarını yapabilmek için de veri işlem dili kullanılmaktadır. Birincil depoda veri işleme dili olarak görülebilen servis dili aslında veri işlem dili fonksiyonları için bir etkinlik çerçevesi sağlamakta, veri işlem dili de böylece veritabanı için bir "arabirim dili" olarak karşımıza çıkmaktadır. Bu dilin komutları ve servis dili bir uygulama programı içinde kaynaşmış durumdadır. Bu iki dil arasındaki fark sadece teorik olup bu dillerin beraber kullanıldığı programlarda, dilleri birbirinden fiziksel olarak ayıran "başlangıç" ve "bitiş" noktaları yoktur.

Veri işlem dilinin veritabanı ile olan bütün arabirimleri, sembolik ve mantıksal seviyededir. Bu yüzden veri işlem dili ile yazılan uygulama programlarının aygıt bağımsızlığı vardır. Bunlara ilaveten programlar aldıkları verileri kullanıcı çalışma alanına yerleştirirler. Diğer bütün fiziksel giriş/çıkış işlemlerinden ve tampon alanlarının yönetiminden DBMS sorumludur. Hiçbir veri işlem dili girişi, bir aygıt veya ortam için referans taşımaz. Dolayısıyla, veritabanı dili ile yazılan şemalar, mantıksal veritabanı tanımlayıcılarıdır. Yine bu nedenle bir veritabanı, DBMS'de tanımlı bir ikincil depolama cihazında saklanabilmektedir. Bu cihazlardan bazıları sıralı erişim özelliklerinden dolayı -manyetik tape üniteleri gibi- veri işlem dilinin bütün özelliklerinden tam olarak yararlanmaya izin vermemektedir²⁵. Meseleyi daha anlaşılır kılmak için bir şekil üzerinde izahını yapalım.



Şekil 1. Kavramsal Veritabanı Yönetim Sistemi

²⁵ Emin D. Aydın , a.g.e., s. 58-66

Numaralandırılmış oklar, kullanıcı programı-1 tarafından yapılmış veri çağrılarının akışını temsil etmektedir. Diğer kullanıcı programları tarafından yapılan çağrılar ise DBMS tarafından karşılanıp sıra ile işleme konulmaktadırlar.

- 1- DBMS'e yapılan bir veri çağrısı. DBMS'e yapılan tüm servis çağrıları veri işlem dilindedir.
- 2- DBMS yapılan çağrıyı analiz eder ve çağrı argümanlarına göre veritabanı şemasından ve çağrıyı yapan kullanıcı programı alt-şemasından gerekli veritabanı bilgisini ortaya çıkarır. Şema, veri karakteristikleri ve veri birimleri arasındaki ilişkiler açısından veritabanı hakkında bir açıklama bilgisi taşımaktadır. Şemanın bir alt kümesi olan alt-şema ise program tarafından kullanılan veritabanı bölümü hakkında bilgi vermektedir. Bu veritabanı bölümü bilgileri DBMS'in izin verdiği yapıdadır ve yine DBMS tarafından programın çalışacağı kullanıcı çalışma alanı içine transfer edilir.
- 3- Çağrının şema ve alt-şemadan aldığı servis ve bilgiler ile birlikte işletilebilmesi için, DBMS işletim sisteminden fiziksel giriş/çıkış işlemleri ister.
- 4- İşletim sistemi ikincil bilgi deposu ile dayanışma halindedir.
- 5- İşletim sistemi, ikincil bilgi depo ve sistem tampon bellekleri arasında veri transferi yapar.
- 6- DBMS çağrının gerektirdiği gibi sistem tampon bellekleri ve programın çalıştığı kullanıcı çalışma alanına veri transferi yapar. İkincil depo ve kullanıcı çalışma alanı arasında gereken herhangi bir başka veri transfer istekleri DBMS tarafından karşılanır.
- 7- DBMS çağrı işletiminin sonunda çağrıyı yapmış program için bir durum raporu oluşturur. Güncel durum bilgisi, hata durum kodları, alan ve kayıt adı bu raporun elemanlarıdır.
- 8- Kullanıcı çalışma alanı içindeki veri, servis dili özelliklerine göre işleme sokulur.
- 9- DBMS sistem tamponlarını yönetir. Sistem tamponları DBMS ile çalışılan tüm programlar tarafından paylaşılır. Kullanıcı programları, tamponlardan DBMS aracılığı ile yararlanır.

1.2.9. Sistem Destek Fonksiyonları

Sistem destek fonksiyonları ile :

- 1- Günlük çalışmalarda veritabanı destek ve servis rutinleri olan aktarım (dump), düzenleme (edit), döküm (print), yükleme, istatistik, analiz, karşılaştırma v.b. rutinleri
- 2- Aktivite raporu, ara nokta (checkpoint) ve geri dönüş (roll-back) özelliklerine sahip bir kurtarma sistemi
- 3- Şema veya alt-şemada değişiklikler yapıldığında bu değişikliklerin veritabanına da yansıtılmasına izin verecek bir dilin varlığı

anlatılmaktadır. Bu dilin yokluğu halinde yapılacak tek şey, bu değişiklikleri taşıyan yeni bir şema oluşturmak ve veritabanını bu yeni şemaya göre yeniden yapılandırmaktır²⁶.

²⁶ Emin D. Aydın , a.g.e., s. 57-63

1.2.10. Verinin Korunması (Data Security)

Bir veritabanı organizasyonun çeşitli verileri, paylaşılır veya paylaşılar. Bu tip paylaşılan veritabanlarında iki tür koruma gereklidir. Birincisi yetkisiz veri erişimine karşı (gizlilik), ikincisi ise veriyi, diğer programların dolaylı etkilemesine karşı (bütünlük).

1.2.11. Veri Yapısı (Data Structure)

Veri yapısından kasıt verinin özellikle erişim (access) yöntemlerine göre düzenlenmesidir. Fiziksel yapı verinin depolama cihazından görüldüğü biçimde, mantıksal yapı ise verinin kullanıcıya görüldüğü biçimde düzenlenmesi olduğunu görmüştük. Çeşitli tipte veri yapıları mevcuttur.

1.2.11.1. Ardışık Yapılar (Sequential Structures)

Bir ardışık yapıda ilk ve son hariç her eleman bir önceki ve bir sonraki eleman ile bağlantılıdır. Bir liste ardışık yapılara iyi bir örnektir. Liste tek veya çift yönlü olabilir.

1.2.11.2. Ağaç Yapılar (Tree Structures)

Hiyerarşik bir yapıya sahip olup bu yapı dahilindeki herhangi bir eleman altındaki n sayıda elemana bağlanırken, üstündeki ancak bir elemana bağlanabilmektedir. Yapının en tepedeki elemanına kök denmektedir. Ağaç yapıları herhangi bir genişlik veya uzunlukta olabilmektedir.

1.2.11.3. Şebeke Yapıları (Network Structures)

Veri yapıları içinde en genel olan şebeke yapısıdır. Şebeke içindeki bir eleman herhangi bir başka elemana bağlı olabilir. Ağaç yapılardan farkı ise şebeke içinde bağlantı açısından herhangi bir sınırlamanın olmamasıdır²⁷.

1.2.12. Veri Bağımsızlığı (Data Independence)

Veritabanı kavramı geliştirilmeden önce yapılan programlar yapılarındaki önemli bir eksiklik yüzünden sonradan pek çok problemler çıkarmaktaydılar. Dinamik bir kuruluştaki sistemin tanımlanması hiçbir zaman tam olarak yapılamamakta, yapılan sadece sistem tanımı ile yola çıkıp kodlamaya geçilmekten ibaretti. İster sistem analizinin eksik yapılmasından ister sistemin kendisinden kaynaklansın pek çok değişiklik söz konusuydu. O zamanlar kullanılan 3. kuşak diller yeterince esnek değildi ve yapılan programlar ağırlıklı olarak veriye bağlıydı. Bir dosyaya yeni bir alan eklenmek istendiğinde veya bir dosyaya erişimin değiştirilmesi istendiğinde ciddi problemler çıkmaktaydı.

1980'lerin başında daha yaygın olarak kullanılmaya başlanan DBMS'ler ve 4. kuşak dilleri bu tür problemlerin çoğu ile baş edebilmektedir. Modern DBMS'lerde ise veri bağımsızlığı üst düzeydedir. Veri bağımsızlığı dosya yapılarındaki ve dosyalara erişim metodlarındaki değişikliklerden, uygulama programlarının etkilenmemesi demektir. Bir verinin saklanma yapısının (fiziksel olarak verinin kaydedildiği şekil) ve erişim yönteminin, verinin kullanıldığı uygulamalardan bağımsız olarak belirlenmesi veya değiştirilebilmesi özelliği olarak da tanımlanmaktadır²⁸.

²⁷ Emin D. Aydın , a.g.e., s. 67-95

²⁸ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 198-199

1.3. Veritabanı Yönetim Sistemi (Database Management System-DBMS)

Birbiri ile alâkalı veri toplulukları ile o veriye erişmek için gerekli bir program setinden meydana gelen DBMS'lerin ilk amacı veritabanında bilgileri saklamak ve ondan yeniden bilgi elde etmede kullanılacak uygun ve verimli bir çevre sağlamaktır. Veri toplulukları genellikle veritabanı olarak gösterilmekte, veritabanı da belli bir organizasyon hakkındaki bilgileri içermektedir.

Veritabanı sistemleri geniş hacimli bilgileri yönetmek üzere dizayn edilmiştir. Veri yönetimi, hem bilgi depolamak için gerekli yapıların tanımını hem de bilgiyi kullanıp değerlendirmek için gerekli olan mekanik hazırlıkları içine almalı, veritabanı sistemi de sistem arızaları ve izinsiz erişimler karşısında veritabanında depolanan bilgilerin güvenliğini sağlamalıdır. Eğer veri değişik kullanıcılar arasında paylaşılmışsa, sistem, oluşacak muhtemel anormal sonuçlardan da korunmak zorundadır. Çoğu organizasyonlarda bilginin önemi sebebiyle veritabanı değerli bir kaynak durumundadır. Bu durum veri yönetiminin etkinliği konusunda çok sayıda kavram ve tekniklerin geliştirilmesine yol açmıştır.

Bir bankanın müşterilerine ait tasarruf hesaplarını göz önüne getirecek olursak tasarruf hesabı ve banka kayıtları, sürekli sistem dosyalarında saklanmaktadır. Bu dosyalara ek olarak da sistemde dosyaların yönetimini sağlayan çok sayıda uygulama programları (alacak-verecek, yeni hesap ekleme, belli bir zaman içerisindeki hesap durumlarının gösterimi v.b.) bulunmaktadır. Uygulama programları, sistem programcıları tarafından bankanın ihtiyacına uygun olarak yazılmakta ve ihtiyaç duyuldukça da sisteme yeni programlar eklenmektedir. Zamanla daha fazla dosya ve uygulama programı sisteme dahil edilecektir. Bu dosya ve programlar uzun bir zaman periyodu içerisinde oluşturuldukları için farklı programlama dillerinde farklı formatlara sahip dosyalar oluşacaktır.

Yukarıda belirtilen ortam geleneksel işletim sistemlerince desteklenen tipik bir dosya işleme sistemidir. Sürekli kayıtlar değişik dosyalarda saklanmakta, kayıtlardan çıkış almak ve uygun dosyalara yeni kayıtlar eklemek için çok sayıda farklı uygulama programları yazılmaktadır. Bu tür projelerin değişik dezavantajları vardır. Bu dezavantajlar DBMS'lerin gelişmesini teşvik etmiştir²⁹.

1-Veri fazlalığı ve düzensizliği : Dosyalar ve uygulama programları farklı programcılar tarafından uzun bir zaman dönemi içerisinde oluşturulduğu için aynı bilgi parçası farklı yerlerde depolanmış olabilir. Mesela belli bir müşterinin adresi, hem banka tasarruf hesabında hem de çek hesaplarını içeren dosyada olabilir. Bu fazlalık potansiyel veri düzensizliği kadar yüksek düzeyde depolama ve erişim maliyetine de yol açmaktadır. Veri düzensizliği deyimi ile aynı verinin değişik kopyalarının uyumsuzluğu kastedilmektedir. Eğer bir müşterinin adresi değişirse ve bu değişiklik de yalnızca tasarruf kayıtlarında yapılırsa veri uyumsuzluğu bariz bir şekilde ortaya çıkacaktır.

²⁹ Henry F. Korth and Abraham Silberschatz, 1986, Database System Concepts, McGraw-Hill Book Company, University of Texas at Austin, ISBN : 0-07-100529-3, p. 1-2

2-Veriye erişimde güçlük : Farzedelim ki banka memurlarından birisi alan kodu 78733 olan şehir bölgesinde bulunan müşterilerinin listesini almak istiyor. Memur, bilgi işlem bölümü ile temasa geçerek böyle bir program hazırlamalarını ister. Bu, sistem dizayn edilirken önceden düşünülmendiğinden olağan dışı bir istek olup böyle bir liste tanzim etmek için elde hiçbir uygulama programı mevcut değildir. Sadece tüm müşterilerin listesini döken bir uygulama programı vardır. Banka memurunun önünde şimdi iki seçenek vardır. Ya tüm müşterilerin listesini alıp sekreterlerden birinin yardımıyla ihtiyaç duyulan bilgileri elle düzeltecek veya bilgi işlem bölümünden böyle bir programın yazılmasını isteyecektir. Her iki alternatif de açıkça tatmin edici değildir. Böyle bir programın yazıldığını, birkaç gün sonrada aynı memurun o listeyi 1.000.000 veya daha fazla miktarda hesap bakiyesine sahip müşterileri dahil ederek düzenleyecek olursa ne olacaktır? Memur yine ikisi de tatmin edici olmayan iki seçenekle karşı karşıyadır. Vurgulamak istediğimiz şey şudur. Böyle bir ortamın hiç kimseye ihtiyaç duyduğu veriye uygun ve verimli bir biçimde elde etmesine imkân vermediğidir.

3-Veri izolasyonu : Veri farklı dosyalara dağıldığı ve dosyalar farklı formatlarda olduğundan uygun veriye elde edebilmek için yeni uygulama programları yazmak da böylesi bir sistemde zor olacaktır.

4-Birden fazla kullanıcının varlığı : Bir çok sistem veriye eşzamanlı olarak güncelleştirirken sistemin performansını artırmak ve daha hızlı bir cevap zamanı elde etmek için çok yönlü kullanıcılara izin vermektedir. Böylesi bir ortamda birbirine uygun güncelleştirmelerin birbirlerine tesiri ile veri uyumsuzluğu meydana gelmektedir. Örneğin müşteri A'nın 5.000.000 TL'lik banka hesabından aynı anda iki farklı kişi 500.000 ve 1.000.000 TL. para çekecek olursa hesap uygun olmayan bir duruma gelecektir. Hesapta 3.500.000 değil ya 4.500.000 veya 4.000.000 TL. bulunacaktır. Bu ihtimale karşı önceden tedbir almak için bazı koruma tedbirlerini yürürlüğe sokmak gerekecektir. Veriye başlangıçta koordine edilmemiş farklı uygulama programları tarafından erişilmeye çalışıldığından böyle bir tedbiri almak çok zor olacaktır.

5-Uyum problemleri : Veritabanında depolanan veri değerleri belli türde uyum sınırlarını korumalıdır. Mesela bir banka hesap bakiyesi hiçbir zaman önceden belirlenmiş bir miktarın -örneğin 25.000.000 TL.- altına inmemelidir. Bu sınırlar sistemde mutlaka uygulanmalıdır. Çeşitli uygulama programlarına uygun kodları eklemek suretiyle böyle bir tedbir alınabilir. Fakat yeni sınırlamalar eklendiğinde onları uygulamaya sokmak için programları değiştirmek zordur³⁰.

1.4. Veri Soyutlama

DBMS'lerin çeşitli kullanıcılara bu dosyalara erişim ve düzenleme imkânı veren birbiri ile ilişkili dosyalar ve programlar seti topluluğu olduğunu yukarıda anlatmıştık. Veritabanı sisteminin temel amacı ise kullanıcılara verinin soyut bir görünüşünü temin etmektir. Yani sistem, verinin nasıl saklanıp muhafaza edildiği hakkındaki detayları gizleyerek veri soyutlamayı (data abstraction) temin etmektedir. Bununla beraber sistemin kullanılabilir olması için, veri etkin bir şekilde yeniden elde edilebilmelidir. Bu etkinlik durumu kompleks veri yapılarının veritabanında temsil edilebilmesini de sağlamaktadır. Veritabanı

³⁰ Korth and Silberschatz, a.g.e., p. 2-3

sistemleri daha ziyade bilgisayar eğitimi olmayan kişilerce kullanıldığı için bu karmaşık yapı veritabanı sistemi kullanıcılarından, veritabanınca görüntülenebilen çeşitli soyutlama seviyeleri tanımlanarak, gizlenmelidir.

1.4.1.1. Fiziki Seviye

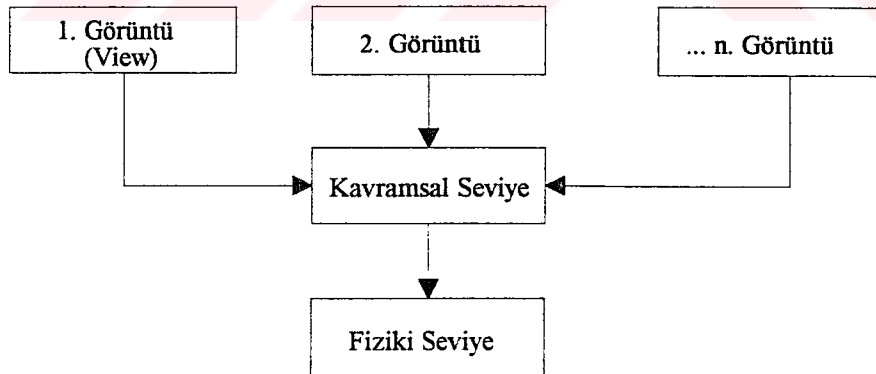
Verinin gerçekte nasıl depolanacağını tanımlandığı en düşük soyutlama seviyesidir. Kompleks ve düşük seviyeli veri yapıları detaylı olarak bu seviyede tanımlanmaktadır.

1.4.1.2. Kavramsal Seviye

Bu soyutlama seviyesinde gerçekte ne tür verilerin veritabanında saklanacağı mutlaka belirtilmekte ve mevcut veriler arasındaki ilişkiler tanımlanmaktadır. Tanımlar az sayıda ilişkili yapılara dayanmaktadır. Kavramsal seviye tanımları basit yapılarına karşın kompleks fiziki seviye yapıları gerektirmekte, bu seviyeden kullanıcıların haberdar olmaları gerekmemektedir. Kavramsal veri soyutlama seviyesi daha ziyade veritabanı yöneticilerince kullanılmaktadır.

1.4.1.3. Görüntü Seviyesi

Bu seviyede tüm veritabanının sadece bir bölümü tanımlanmaktadır. Dolayısıyla en yüksek seviyedeki veri soyutlamayı ifade etmektedir. Kavramsal seviyenin basitleştirici yapısına rağmen, ortada veritabanının büyüklüğünden kaynaklanan kompleks bir yapı bulunmaktadır. Veritabanı sistemini kullanan çoğu kullanıcı tüm bu bilgilerle ilgilenmemekte, bunun yerine sadece veritabanının belli bir bölümüne ihtiyaç duymaktadırlar. Bu kullanıcıların sistemle etkileşimi, soyutlamanın görüntü seviyesinde tanımlanmakta, diğer görüntüler ise sistem tarafından sağlanmaktadır. Soyutlamanın üç seviyesi ve aralarındaki ilişkiler aşağıdaki gösterilmiştir³¹.



Şekil 2. Veri Soyutlamanın 3 Seviyesi

Örneğin PL/1 için bu seviyeler aşağıdaki gibi tanımlanmaktadır³².

³¹ Korth and Silberschatz, a.g.e., p. 4-5

³² C. J. Date, November 1987, An Introduction to Database Systems, Fourth edition, Addison-Wesley Publishing Company, Volume 1, U.S.A., ISBN : 0-201-14201-5 (v.1), p. 31

Görüntü Seviyesinde :

DCL 1 EMPP,
 2 EMP# CHAR (6),
 2 SAL FIXED BIN (31);

Kavramsal Seviyede :

Employee

Employee_number	character (6)
department_number	character (6)
salary	numeric (5)

Fiziki Seviyede :

prefix	type=byte(6), offset=0
emp#	type=byte(6), offset=6, Index=emp#
dept#	type=byte(4), offset=12
pay	type=fullword, offset=16

Soyut seviyeler arasındaki farklılığı göstermek için programlama dillerindeki veri tiplerinin muhtevası ile bir karşılaştırma yapılabilir. Yüksek seviyeli programlama dillerinin çoğu kayıt kavramını desteklemekte, örneğin Pascal gibi dillerde bir kayıt aşağıdaki gibi tanımlanmaktadır³³.

```

type musteril = record ;
    isim : string ;
    cadde : string ;
    sehir : string ;
end ;

```

Bu örnekteki kayıt müşteri isminde olup isim, cadde ve şehir alanlarına sahiptir. Her bir sahanın ismi ve tipi arasında yakın bir ilişki vardır. Hesap numarası ve kalan parayı gösteren hesap kaydı, isim ve ücret içeren çalışan kaydı bu konuda değişik örneklerdir. Fiziki seviyede bir müşteri, hesap veya çalışan kaydı, depolama alanlarının arkasında yer alan bayt (byte) veya kelime (word) blokları olarak tanımlanmaktadır. Müşteri kayıt tipi örneğimize dönecek olursak müşteri kaydı herhangi bir değişkeni bildirmemektedir. Çünkü Pascal gibi dillerde böyle değişkenleri tanımlamak için şu ifade kullanılmaktadır.

```

var musteril : musteril ;

```

Bu ifade ile müşteri değişkeni şu anda müşteri kayıt tipini gösteren bir depolama alanıdır. Veritabanı şeması kavramı programlama dillerinin tip tanımlama (type definition) tasarımlarına benzemektedir. Belirtilen tipteki bir değişken belirli bir zamanda özel bir değere sahiptir. Dolayısıyla programlama dillerindeki değişken değeri kavramı, veritabanı şeması örneğine tekabül etmektedir. Veritabanılarında çeşitli şemalar mevcut olup veri soyutlama seviyelerini izleyecek şekilde bölümlere ayrılmıştır. En düşük seviye fiziksel

³³ Korth and Silberschatz, a.g.e., p. 5

şema, orta seviyede kavramsal şema, en üst seviyede ise bir alt-şema vardır. Genellikle veritabanı sistemleri bir fiziki, bir kavramsal ve birkaç alt-şemayı desteklemektedir.

1.5. Veri Modelleri

Veritabanı yapısını tasvir edebilmek için veri modeli kavramını tanımlamak gereklidir. Yani veritabanının yapısını tanımlamak için önce veri modeli tanımlanmalıdır. Veri modeli; verinin sınırlarını, anlamlarını, ilişkilerini ve tanımlarını sağlayan fikri araçlar topluluğudur. Çok sayıda veri modelleri planlanmıştır. Bunlar üç farklı grupta mütalaa edilmektedirler³⁴.

1.5.1. Nesne Tabanlı Mantıksal Modeller (Object-Based Logical Models)

Bu modeller kavramsal ve görüntü seviyesindeki verileri tanımlarken kullanılmaktadır. Hayli esnek yapısal kapasiteleri ile veri yapısını kesin olarak tanıtmaktadırlar. En az 40 farklı modele sahiptirler. Yaygın olarak bilinen bazı modelleri şunlardır.

- a) Varlık-İlişki Modeli (The Entity-Relationship Model veya E-R Model)
- b) İkili Model (The Binary Model)
- c) Anlamsal Veri Modeli (The Semantic Data Model)
- d) Mantıksal Bilgi Modeli (The Infological Model)

İlerleyen bölümde nesne temelli mantıksal modelleri temsilen Varlık-İlişki modelini ele alacağız. Bu modeli seçmemizin sebebi geniş bir kullanım alanına sahip olması ve veritabanları için en uygun veri modeli olarak kabul edilmesidir.

1.5.2. Kayıt Temelli Mantıksal Modeller (Record-Based Logical Models)

Nesne tabanlı mantıksal modeller gibi kavramsal ve görüntü seviyesindeki verileri tanımlamada kullanılmakla beraber daha yüksek seviyeli tanımlar da belirtilebilmektedir. Veri yapısının tam manasıyla tanımlanması için çeşitli imkânlarla sahiptirler. Geniş kabul görmüş üç büyük veri modeli vardır.

- a) İlişkisel Model (Relational Model)
- b) Şebeke Modeli (Network Model)
- c) Sıralı Model (Hierarchical Model)

1.5.3. Fiziksel Veri Modelleri (Physical Data Models)

Bu veri modelleri en düşük seviyedeki verileri tanımlamak için kullanılmaktadır. Mantıksal veri modellerinin çokluğuna karşın pek az sayıda fiziksel veri modeli vardır. En çok bilinenleri şunlardır.

- a) Birleştirilmiş Model (Unifying Model)
- b) Çerçeve Modeli (Frame Model)

1.6. Veri Bağımsızlığı

Bir veritabanının görüntüleyebileceği herhangi bir soyutlama seviyesindeki şema tanımlarını, daha yüksek bir seviyedeki şema tanımının etkisi olmadan biçimlendirme

³⁴ Korth and Silberschatz, a.g.e., p. 6-9

kabiliyeti veri bağımsızlığı (data independence) olarak adlandırılmaktadır. İki tür veri bağımsızlığı bulunmaktadır.

1.6.1. Fiziksel Veri Bağımsızlığı

Dosyalarda bulunan kayıtlara erişim için farklı metotlar bulunmaktadır. Bu metotlardan birisi de erişimleri oldukça hızlandıran indeksleme sistemleridir. Fakat bu indeksler diskte belirli bir yer işgal etmekte, ayrıca bozulma tehlikesini de içermektedirler. Verilerin muhtevasına bağlı olarak da indeksleme sistemleri arasından da seçim yapılması gerekmektedir. Uygulamanın başında bir erişim sistemi en iyisi olarak hesaplanıp seçim yapılmış bile olsa daha sonra değişen veri muhtevasına bağlı olarak farklı bir erişim metodu da gerekebilmektedir. İşte bu noktada erişim metodunu değiştirirken mevcut programda bir değişim gerekmemesi oldukça önemli bir esnekliktir. Bu esneklik fiziksel veri bağımsızlığı olarak tanımlanmaktadır. Diğer bir deyişle programların erişim metodlarından bağımsız olarak inşa edilebilmesi şeklidir.

1.6.2. Mantıksal Veri Bağımsızlığı

Programlama çalışmalarının başlangıcında yapılan sistem çözümlemesi zaman zaman sistemde yapılan değişiklikler yüzünden geçersiz olmaktadır. Bazı yeni dosyaların eklenmesi, bazı dosyaların iptali, dosyalar arasındaki ilişkilerin değiştirilmesi, dosyalara yeni alanların eklenmesi, bazı alanların iptali veya alanların uzunluk, tip v.b. özelliklerinin değişmesi söz konusu olabilmektedir. Bu tür değişiklikler programlarda da bazı önemli değişimlere sebep olmaktadır. Modern DBMS'lerdeki diller ile hazırlanan programlarda ise bu gerekli değişim minimuma indirilmekte ve büyük bir esneklik sağlanmaktadır. İşte mantıksal veri bağımsızlığı bu esnekliğin adıdır.

Bu esneklik uygulama programlarının yeniden yazılmasına ihtiyaç bırakmadan kavramsal şemaların biçimlendirilmesi özelliği olarak da bilinmektedir. Kavramsal seviyedeki değişiklikler, veritabanının mantıksal yapısı değiştirildiği zamanlarda gereklidir. Uygulama programları veriye erişebilmek için verinin mantıksal yapısına sıkı sıkıya bağlı olduğundan, veri bağımsızlığına ulaşmak fizikî veri bağımsızlığına ulaşmaktan daha zordur. Veri bağımsızlığı kavramı modern programlama dillerindeki soyut veri tiplerine bir çok yönden benzemektedir. Hem veri bağımsızlığına hem de veri soyutlamaya ait detaylar kullanıcılardan gizlenmektedir. Böylece kullanıcıların dikkati, düşük seviye gerektiren detaylardan ziyade genel yapı üzerinde yoğunlaşmaktadır³⁵.

Modern DBMS'ler veri bağımsızlığı sağlamak için özel bir mimari ile üç katlı olarak inşa edilmektedir. Bu modüller birbirleri ile haberleşebilmekte fakat bağımsız bir şekilde hareket etmektedirler. Bir dosya için her katta bilgiler bulunmaktadır. Üç seviyeli yaklaşım oldukça kabul görmüş bir yaklaşımdır. Çünkü üç seviyeli yapı hem fiziksel hem de mantıksal veri bağımsızlığını sağlamaktadır. 1975 yılında American Ulusal Standartlar Enstitüsü'nün (ANSI) Standartlar ve Planlama Komitesinin (SPARC) kararları ile onaylanmış ve bir standart haline gelmiştir. Fiziksel, mantıksal ve dış katmanlar üç katlı veritabanını oluşturan elemanlardır. Dosyalar içindeki veriler gerçekte sadece fiziksel katta bulunmaktadır. Diğer katlarda ise dosyalar hayali (virtual) olarak bulunmaktadır.

³⁵ Korth and Silberschatz, a.g.e., p. 10

Mantıksal ve dış katmanlarda ise dosya yapıları ve ilişkileri ile genel bilgiler bulunmaktadır.

1.6.2.1. Fiziksel Kat ve Fiziksel Dosyalar

Bu dosyalar basitçe COBOL, PL/I, C, PASCAL gibi dillerdeki read, write, rewrite benzeri komutlarla işlenen dosyalar olarak bilinir. Klasik veri işlemde kullanılan bütün dosyalar bu tür dosyalardır. Bu dosyalarda kullanıcı için anlamlı olmayan ama yapısal olarak gerekli bir takım bağlantı alanları söz konusudur.

Tablo 1. Depo isimli fiziksel dosya

Depo no	Şehir	Hacim	Iptal
101	İstanbul	13500	
102	İzmir	15000	
103	Ankara	13500	
104	Adana	12000	*
105	Kayseri	11300	

Yukarıdaki tablo, depo fiziksel dosyasında 104 nolu depo kaydının iptal edildiğini göstermektedir. Depo fiziksel dosyasında kullanıcılar için çok fazla bir şey ifade etmeyen bir alan - iptal alanı- görülmektedir. Bir kayıt iptal edildiğinde gerçekte çok zaman tutan bir işlem gerekmektedir. Bu uzun süren işlem yerine ilgili kaydın iptal edildiğine dair bir bilgi (*) iptal alanına yerleştirilmekte ve DBMS sanki o kayıt yokmuşcasına hareket etmektedir. Bu işleme mantıksal iptal denmektedir. Daha sonra bilgisayarın az yüklü olduğu herhangi bir zamanda tüm (*) işaretli kayıtlar fiziksel olarak yok edilmektedir. Böylece toplam iptal süresi daha az zaman almakta ve darboğazlar oluşmamaktadır³⁶.

Veritabanlarının tasarımında mantıksal dosyalar kadar fiziksel dosyalar da çok önemlidir. Fiziksel dosyaların tasarımı programların çalışma performansını oldukça etkilemektedir. Çünkü fiziki yapılar erişim metotlarını belirlemektedir. Yani gerçekte fiziksel yapıyı seçmek, erişim metodunu da seçmek demektir. Fiziksel dosyalar 4. kuşak dillerde fiziksel şemada, 3. kuşak veya dBASE sistemlerinde ise programın içine gömülü (embedded) haldedir. Bu tür dosyalar basit dosyalardan biraz daha fazla yer kaplamaktadır ve programlanmaları daha zordur. Verilerin bozulması söz konusudur, fakat daha hızlı işlem yapabilmesi açısından bir avantajdır. Eğer esnek bir alt yapı hazırlanmamış ise daha farklı erişim sistemlerine dönüşüm sağlamak zor olacaktır. Bu problem veri bağımsızlığı yüksek DBMS'ler kullanılarak büyük oranda çözülebilmektedir. Dosyaların fiziksel düzenlenişlerini 2 grupta incelemek mümkündür:

1- Fiziksel olarak sürekli dosyalar : Fiziksel olarak sürekli dosyalar bir manyetik ortamda belirli bir adresten başlamakta ve bütün bilgiler bir ipteki tesbih taneleri gibi ardarda dizilmektedir.

³⁶ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 199-201

2- Fiziksel olarak dağıtılmış dosyalar : Fiziksel olarak dağıtılmış dosyalarda ise bu süreklilik değişik sebeplerden dolayı kesintiye uğramakta ve bilgilerin devamı farklı bir adresten başlamaktadır.

Bu ayrımın bir şekilde giderilmesi, mantıksal görünümde sürekliliğin sağlanması gerekmektedir. Bunun için bir tür köprü görevi yapan göstergeler (pointers) kullanılmaktadır. Dağıtılmış veri bloklarının sonunda bir sonraki veri bloğunun başlangıç adresini veren gösterge adında bir alan vardır. Göstergeler kullanılarak farklı amaç ve etkinlikte veri yapıları inşa edilebilmektedir. Burada incelenecek olan yapılar şunlardır:

1.6.2.1.1. Tek Yönlü Bağlı Listeler

Bu yapılarda her kayıt birbirinden bağımsız olarak düşünülmektedir. Her kaydın sonunda bir sonraki kaydın kayıt numarasını veren bir pointer alanı bulunmaktadır. Ayrıca her bağlı listenin ilk kaydını (mantıksal olarak) belirten bir bilgi bulunmaktadır. Bu başlangıç adresi "s" olarak bilinir. Son kaydın pointer alanı ise -son kayıt olduğu anlaşılсын diye- özel bir değer olarak -örneğin 0- belirtilir. Tek yönlü listelerde sadece bir sonraki kayda gidilebilir. Bir önceki kayda direkt erişmek mümkün değildir. Bir önceki kayda erişmek için listenin başınsan başlayıp aynı noktaya varıncaya değin devam etmek gereklidir.

1-Kayıt iptali : Bağlı listelerde kayıt iptali diğer dosyalardaki kayıt iptal işleminden daha farklı olarak yapılmaktadır. İptal edilen kayıt fiziksel olarak yok olmamakta sadece o kaydı gösteren pointer bir sonraki kaydı göstermektedir. Böylece iptal edilen kaydı gösteren herhangi bir kayıt kalmamakta ve bu kayda erişim mümkün olmamaktadır³⁷.

2-Kayıt ekleme : Kayıt ekleme işleminde havuzdan boş bir kayıt alınmakta, bilgiler alanlara yerleştirilmekte, daha sonra da göstergeler ayarlanmaktadır. Eklenmesi istenen noktaya bağlı olarak bir önceki kayıt (mantıksal olarak) eklenen kaydı, eklenen kayıt da bir sonrakini göstermektedir.

3-Arama ve listeleme : Listeleme işlemi normal dosya veya dizilerden daha farklıdır. Normal bir dosyayı listelemek için önce ilk kayda daha sonra 2. 3. kayıtlara gidilmektedir. Kayıtlara erişim böylece dosyanın sonuna kadar kayıt numaralarının birer birer arttırılması ile sağlanmaktadır. Fakat bağlı listelerde ise başlangıç kaydına gidilmektedir (s). Başlangıç kaydı 1 nolu kayıt olmayabilir. Daha sonra söz konusu kaydın gösterge alanında bulunan adresteki kayda gidilir. Pointer alanında 0 değeri oluncaya kadar bu işlem sürer. Arama işleminde ise aranan kayda rastlanırsa bir sonraki kayda gidilmez ve işlem durdurulur. Arama işlemleri sıralı olmak zorundadır, daha fazla kısaltılamaz. Dolayısıyla n adet kayıt olan bir listede en kötü durumda n adet işlem gerekecektir. Averaj olarak n/2 işlem gerekecektir. Halbuki n kayıt için daha az işlem gerektiren erişim metodları söz konusudur.

Gerçekte bağlı listelerde 2 ayrı liste vardır. Verilerin listesi ile boş ve kullanılmayan kayıtların oluşturduğu havuz adı verilen liste. Havuz bağlı listelerde çok önemli bir kavramdır. İptal edilen kayıtlar birbirlerine pointerler ile bağlanmakta daha sonra yeni bir kayıt eklenmesi gerektiğinde havuzdan boş bir kayıt alınıp ve kullanılmaktadır. Hem veri listesinin (s), hem de havuz listesinin de bir başlangıcı (v) bulunmaktadır.

³⁷ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 232-234

1.6.2.1.2. Çift Yönlü Bağlı Liste

Tek yönlü listelerdeki bir pointer alanına karşılık çift yönlü bağlı listelerde 2 adet pointer vardır. İlk pointer bir sonraki kaydı gösterirken 2. pointer bir önceki kaydı göstermektedir. Çift yönlü listelerde tek bir havuz bulunmaktadır ve listeleme, arama işlemleri tek yönlü listelerdeki gibidir. Tek fark bir önceki kayda doğrudan erişimin mümkün olmasıdır.

1.6.2.1.3. Ağaç Yapıları

Ağaç yapıları çok özel bir düzenlemeye sahiptir. Pek çok farklı ağaç yapısı bulunmaktadır. Burada sadece ikili arama ağacı incelenecektir. Aslında bu yapılar bir fiziksel kayıt içinde, m adet mantıksal kaydın ve $m+1$ çocuğu olan ağaçlar için geçerlidir. Bu ağaçlara m yönlü arama ağaçları ismi verilmektedir. Arama ağacı bir tür indekstir ve veri kayıtlarına hızlı erişmek için kullanılmaktadır. Sıralı aramaya göre büyük avantajlara sahiptirler. Arama ağacının kayıt deseni farklıdır. Kayıta aranacak anahtar alanın verisi ve ayrıca 2 adet pointer bulunmaktadır. Sol ve sağ çocuk pointeri³⁸.

Her kaydın en fazla 2 çocuğu bulunabilmektedir. Çocukların adresleri pointer alanlarında saklanmaktadır. Ağaç yapısının en tepesindeki kayda kök kayıt, hiç çocuğu olmayan kayda ise uç kayıt veya yaprak kayıt olarak denilmektedir. Ağaç yapısı belirli bir düzende oluşturulmaktadır. Bir kaydın sol tarafından bağlanan çocuk kayıt ve onun da çocuklarından oluşan kayıtlar sol alt ağacı temsil ederken, sağ taraftan bağlanan kayıt ve onun çocuklarından oluşan kayıtlar da sağ alt ağacı temsil etmektedir. En temel özelliklerini ise şu şekilde özetlemek mümkündür: Bir kaydın sol alt ağacının içindeki her veri, anne kaydın verisinden daha küçük, benzer şekilde bir kaydın sağ alt ağacındaki her veri de anne kaydın verisinden daha büyüktür.

1-Kayıt arama : Kayıt arama işlemi kök kayıttan başlamaktadır. Eğer aranan değer kayıt değerinden küçük ise sol taraftaki kayda dallanılır. Tam tersi durumda ise arama sağ taraftaki kayıttan devam eder. Arama kayıt bulununcaya veya bir uç kayda varıncaya kadar devam eder. Aramanın adedini belirleyen ağacın derinliğidir. Oluşan ağacın dengeli olması ağacın derinliğini azaltır. Böylece arama süresi kısılır. Bazı ağaçlar sıralı aramalardaki kadar kötü sonuç verecek şekilde dengesiz olmuşlardır.

2-Kayıt ekleme : Eklenen her kayıt sıralı olma özelliğini bozmayacak şekilde yerleştirilmektedir. Kayıt ekleme öncesi, eklenecek kaydın yerini bulmak üzere bir uç kayda kadar arama benzeri bir işlem yapılmaktadır. Her yeni kayıt, uç kayıt olacak şekilde yerleştirilmektedir. Aynı bilgi kümesinin farklı sıralar ile ağaca eklemesi halinde farklı sıralar ortaya çıkacağı da unutulmamalıdır. Ağaç yapısını içinde tutan dosyadan, gerçek verilerin bulunduğu dosyaya bir bağlantı bulunmaktadır. İndeks dosyasına ise bir pointer alanı eklenmiş haldedir. Bu pointer anahtar bilginin veri dosyasındaki karşılık geldiği kayda bağlanmaktadır.

1.6.2.1.4. İkiz Çocuk Pointer Yapısı

Bu sistemde iki dosya arasındaki ilişki 1:n yani hiyerarşik bir ilişki olması gerekmektedir. Dolayısıyla kayıtlara daha hızlı bir erişim sağlanmaktadır. İlk arama yapıldıktan sonra

³⁸ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 235-237

eğer varsa ikiz pointerler vasıtasıyla diğer kayıtlara da sıralı olarak erişmek mümkün olmaktadır³⁹.

1.6.2.2. Mantıksal Kat ve Mantıksal Dosyalar

Bu dosyalar gerçekte fiziksel olarak var olmayan fakat fiziksel dosyalardan üretilen dosyalar olup sadece kullanıcıyı ilgilendiren bilgi alanları mevcuttur.

1.6.2.3. Dış Kat ve Dış Dosyalar

Dış dosyalar, bir mantıksal dosyanın alt kümesi veya birbiri ile ilişkili birden çok mantıksal dosyanın bir kombinasyonu olabilmektedirler.

1.7. Veritabanı Tasarımı

Veritabanlarının oluşturulmasında, mantıksal dosyalarının tasarımı oldukça önemli bir aşamadır. Dosyalar öyle tasarlanmalıdır ki hem gerçeği eksiksiz ve doğru olarak yansıtmalı hem de programlanması kolay olmalıdır. Bilgilerin doğru ve tutarlı kalmasında dosya tasarımının katkısı çok büyüktür. Veritabanının oluşturulması aşamaları aşağıda sıralanmıştır:

- 1- Kuruluşun bilgi ihtiyacının analizi
- 2- Kullanıcıların bakış açılarının belirlenmesi ve buna bağlı olarak dış dosyaların tesbiti
- 3- Belirlenen bakış açılarının bir bütün olarak ele alınıp birbirleri ile olan ilişkilerinin tesbiti ve bir araya getirme çabası
- 4- Dosyaların fonksiyonel bağımlılıklarının tesbiti ve 4. Normal Form'a ininceye kadar parçalara ayrılması
- 5- Entegre olan bilgilerden mantıksal şemaların üretilmesi
- 6- Kapasite, hız gibi faktörleri düşünerek optimal fiziksel yapı ve erişim metotlarının belirlenmesi
- 7- Oluşan mantıksal şemaya bağlı olarak ilk elde gerekli bir takım dış şemaların oluşturulması
- 8- Veritabanı Yönetim Sisteminin şema dilleri veya programları ile şemalarının hazırlanıp uygulama programlarının yazılmasına başlanması.

1.8. Doğruluğun Korunması

Veritabanında biriken bilgilerin mümkün olduğunca yanlışsız ve tutarlı olmasını sağlamak için bir takım önlemler almak gerekmektedir. Bu tedbirler veritabanının tasarımı yapılırken düşünülür ve veritabanı yönetim sisteminin imkânları sayesinde merkezi olarak uygulanır. Herşey bilgi girişi sırasında kontrol edilmeli, hatalı bilgiler anında yakalanmalı ve kullanıcı yazılı bir mesajla uyarılmalıdır. Bunun için :

- 1- En temel doğruluk tedbiri, dosyaların kurallara uygun şekilde tasarımıdır. İleride detaylı olarak açıklanacak olan 4. Normal Form'da olmaları sağlanmalıdır.
- 2- Anahtar alanlarda tekrarlar olmamalı, düzeltilmesine veya boş bilgi girilmesine izin verilmemelidir.

³⁹ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 237-239

3- Bazı alanlar için en büyük/küçük değerler söz konusudur. Örneğin bir depo hacmi hiçbir zaman negatif bir değere sahip olmamalıdır. Dolayısıyla depo hacmi alanı için en küçük değer 0 olarak tanımlanmalıdır.

4- Bazı alanların, mantıksal olarak sahip olması mümkün olan değerleri sabit olabilir. Örneğin malzemelerin renkleri belirli bir kümenin elemanlarından -sarı, mavi, gri- birisi olabilir. Bu tip alanlarda genellikle kullanıcının bilgi girmesi yerine o alana gelindiği zaman bir pencere içinde kullanıcıya sunulan seçeneklerden birisinin seçilmesi sağlanıp kullanıcının daha iyi yönlendirilmesi sağlanmış olur.

5- Bazı alanların alabileceği değerlerin başka bir dosyanın herhangi bir alanında saklı değerler içinde mevcut olması istenebilir. Örneğin personel dosyasındaki depo_no alanına girilebilecek herhangi bir bilgi depo dosyasının depo_no alanı içinde var olmalıdır. Eğer bir personelin depo numarası gerçekte var olmayan bir depo numarası olacaksa bu çok anlamsız olacaktır.

6- Bazı şartları sağlamayan kayıtların iptalinin engellenmesi gerekebilir. Örneğin depo dosyasındaki bir kayıt, bu depoda çalıştığı bilinen personel mevcutsa silme işlemi engellenmelidir.

7- Alanlar için genellikle formatlar tanımlanıp, ne tip değerler alabileceği de belirlenebilmelidir.

1.9. Verilerin İşlenmesi

Veritabanlarını pekçok farklı programlama dili ile işlemek mümkündür. Ayrıca piyasaya ticari olarak sürülen pek çok veritabanının, programlama imkânları da bulunmaktadır. 3. kuşak diller arasında sayılan Cobol, PL/I, Fortran, C gibi diller çokca kullanılmaktadır. Bunun yanı sıra SQL/QUEL/PROLOG gibi 4. kuşak programlama dilleri de hızla yaygınlık kazanmaktadır. 3. kuşak ile 4. kuşak arasında yer alan dBASE-III oldukça tutulmuştur. Dbase'in pek çok farklı türevi zaman içerisinde piyasaya girmiştir. Ashton Tate firmasının dBASE-III+, Borland'ın dBASE IV ve dBASE for Windows yaygın kullanılan versiyonlarıdır.

Nontucket firmasının geliştirdiği dBase sisteminin taklidi Clipper sistemi ise komut seti bakımından en kapsamlı sistem olarak görülmektedir. Clipper'in derlendikten sonra kullanılıyor olması programların çalışmasını olumlu yönde etkilemektedir. Clipper'in sahip olduğu bir avantaj da Assembler ve C ile yazılan alt programları kullanabilmesidir. Unix ortamında çalışan tek dBase taklidi sistem FoxBase+'dır. Bu arada piyasadaki en büyük veritabanı sistemlerinden ORACLE kendi dilini devreye sokmuştur. Programlama dillerini 3 önemli grupta incelemek mümkündür:

1- Yapısal programlama dilleri (procedural) : Bu dillerde yapılması istenilen iş küçük parçalara ayrılır ve her bir iş komuta çevrilip kodlanır. Her yapılması istenilen şey bu dillerde detaylı olarak ifade edilir. (Cobol, Fortran, C, Basic, Pascal, PL/I). Ne tür veriye ihtiyaç duyulacağı ve nasıl elde edileceği kullanıcı tarafından tanımlanan dillerdir.

2- Yapısal olmayan programlama dilleri (nonprocedural) : Yapısal olmayan dillerde neyin nasıl yapılacağına dair bir tarif yoktur. Ne istendiği tarif edilmekte, nasıl çözüme

gidileceği sistem tarafından yapılmaktadır. Bu dillere örnek olarak Sql, Quel, Prolog verilebilir.

3- Yarı yapısal diller : Yapısal ve yapısal olmayan diller arasında bir grup olup yapısal dillerde pek çok komut ile gerçekleştirilen bir iş bu dillerde tek komuta indirgenmektedir. Örnek olarak dBASE, Natural, İlişkisel Cebir dilleri verilebilir⁴⁰.

Yapısal olmayan DML'lerin öğrenilmesi ve kullanımı yapısal olanlara göre daha basittir. Fakat kullanıcılar veriyi nasıl elde edeceklerini belirlemeye yetkili olmadıkları için, elde edilen derlenmiş kodlar yapısal diller tarafından üretilen kodlar kadar etkili değildir. Bu problem çeşitli optimizasyon teknikleri kullanılarak giderilebilmektedir.

Sorgu (query) ise bir bilgiyi çağırma isteğinin ifadesidir. Bilgi almayı sağlayan sorgulama dili (query language) ise yine DML'in bir bölümüdür. Teknik olarak yanlış olsa da sorgulama ve veri yönetim dili pratikte yaygın olarak aynı anlamda kullanılmaktadır⁴¹.

Veritabanları tipik olarak geniş miktarda depolama alanları gerektirmektedir. Bilgisayarların ana hafızaları bu kadar bilgiyi depolayamadığı için veriler disklerde saklanmak zorundadır. Veri, ihtiyaç duyulduğu anda disk deposundan ana hafızaya taşınır. Verinin diske/diskten taşınması, bilgisayarların merkezi işlem ünitesine (Central Processing Unit-CPU) göre yavaş olduğundan diskle ana hafıza arasındaki taşınacak bilgi miktarını minimum seviyede tutacak bir veritabanı sistem yapısı gerekmektedir.

Veritabanı sisteminin amacı veriye erişim imkânını basitleştirmektir. Yüksek seviyedeki görüntüler bunu başarmak için birer yardımcılardır. Sistemin kullanıcıları fiziki detaylarla gerekmedikçe ilgilenmezler. Yine de bir veritabanının performansı ve kullanıcıların hoşnutluğu veya bunların eksikliği önemli bir faktördür. Bir sorgu için cevap elde etme süresi çok uzunsa bu sistemin değerini düşürmektedir. Sistemin performansı veritabanında veriyi göstermek için kullanılan veri yapısının etkinliğine ve bu yapılar üzerinde sistemin en etkin bir biçimde nasıl iş yapacağına da bağlı bulunmaktadır. Veritabanı yöneticisi (database manager) veritabanındaki düşük seviyedeki veri bankası ile sisteme iletilen sorgularla uygulama programları arasında bir arabirim (interface) görevi yapan program modülüdür ve aşağıdaki görevleri yerine getirmekle yükümlüdür⁴².

1-Dosya yöneticisi (file manager) ile karşılıklı etkileşimde bulunmak : İşlenmemiş veri, geleneksel işletim sistemlerince sağlanan dosya sistemini kullanan disklerde depolanmaktadır. Veritabanı yöneticisi çeşitli DML ifadelerini düşük seviyeli dosya sistemi komutlarına çevirmekte, böylece veritabanındaki verinin gerçek olarak depolanmasını, yeniden ele alınıp güncelleştirilmesini sağlamaktadır.

2-Uygulama bütünlüğü sağlamak : Veritabanında depolanan veri değerleri değişik türdeki uygulamalarla güvenilir bir hale getirilmelidir. Bu sınırlar açık bir şekilde veritabanı

⁴⁰ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 211-213

⁴¹ Korth and Silberschatz, a.g.e., p. 11-12

⁴² Korth and Silberschatz, a.g.e., p. 12

yöneticisince belirtilmelidir. Eğer bu tür sınırlar belirtilirse veritabanı yöneticisi bu sınırların herhangi bir şekilde ihlali konusunda veritabanı sonucunun güncelleştirilip güncelleştirilmediğini kontrol edebilecek ve eğer herhangi bir ihlal varsa gerekli tedbirleri alabilecektir.

3-Güvenlik tedbirlerini almak : Bir veritabanı kullanıcısı, tüm veritabanı muhtevasına ulaşmak veya bu muhtevayı bilmek zorunda değildir. Güvenlik tedbirlerini uygulamak da veritabanı yöneticisinin vazifesidir.

4-Yedekleme ve geri yükleme rutinlerini çalıştırmak : Bir bilgisayar sistemi diğer mekanik veya elektriksel araçlara gibi bozulabilmektedir. Bu tür arızaların (disk çökmeleri, güç kaynağı problemleri, yazılım hataları v.b. gibi) değişik sebepleri vardır. Arızaların tesbit edip veritabanını eski haline getirme sorumluluğu veritabanı yöneticisine aittir. Genellikle bu işlemler yedekleme ve geri yükleme işlemlerini başlatmak suretiyle yapılmaktadır.

5-Eşzamanlı kontrol sağlamak : Değişik kullanıcılar, veritabanını aynı anda güncelleştirdiğinde veri uyumluluğu korunamamaktadır. Sistemin, eşzamanlı kullanıcılar arasındaki karşılıklı faaliyetleri kontrol etmesi gerekmektedir. Böyle bir kontrolü sağlamak da veritabanı yöneticisinin görevlerinden birisidir. Veritabanı yönetim sistemine sahip olmanın başlıca sebeplerinden biri hem veriyi hem de o veriye ulaşabilen programların merkezi kontrolünü sağlamaktır. Sistem üzerinde böyle bir merkezi kontrole sahip olan kişi, veritabanı vekili (Database Administrator-DBA) olarak adlandırılır. Veritabanı vekilinin görevleri ise şunlardır.

- 1- Şemaları tanımlayarak orijinal veritabanı projesini oluşturmak
- 2- Depolama yapısı ve erişim metodlarının tanımlamak
- 3- Şema ve fiziki organizasyon değişikliklerini - hem veritabanı şeması değişikliği hem de fiziki depolama organizasyonunun tanımlarını - yapmak
- 4- Veri erişimine izin vermek
- 5- Bütünlük sınırlarını belirlemek

1.10. Veritabanı Kullanıcıları

Bir veritabanı sistemine ilişkin dört farklı kullanıcı türü vardır ve bunlar sistemle etkileşimlerine göre gruplandırılmıştır⁴³.

1-Uygulama programcıları : Bu kişiler bilgisayar profesyonelleri olup, sistemle etkileşimlerini bir ana programlama dili (Cobol, PL/I, Pascal, C gibi) içerisine yazılan (embedded) DML çağrılarını ile yapmaktadırlar. Bu programlara genellikle uygulama programları olarak bakılabilir. DML ifadeleri ana programlama dilinin yazımından oldukça farklı olduğu için DML ön derleyicisi, precompiler olarak isimlendirilen özel bir ön derleyici tarafından -genellikle de # veya \$ gibi özel bir karakter ile belirtilen- DML çağrılarının çevrilirler. Yani DML önderleyicisi DML ifadelerini ana (host) programdaki normal çağrı (call) işlemlerine dönüştürmektedir. Sonuçta, program uygun amaç kodu (object code) üreten ana programa ait derleyici içinden çalıştırılmaktadır. Veritabanının

⁴³ Korth and Silberschatz, a.g.e., p. 12-14

idaresi için Pascal gibi dillerin yapılarını birleştirip kontrol eden özel tipte programlama dilleri de vardır. Pascal R, Focus, Nomad gibi.

2-Günlük kullanıcılar : Program yazmaksızın sistemle etkileşimde bulunan bu kullanıcıların çalışma yöntemi; isteklerini, sorgularını yazarak veritabanı sorgulama diline iletmektir. Fonksiyonu, bir DML ifadesini alıp veritabanı yöneticisinin anlayacağı komutlara çevirmek olan böyle herbir sorgulama bir sorgulama işlemcisine iletilmektedir.

3-Deneyimsiz kullanıcılar : Sistemle etkileşimlerini, uygulama programlarını çağırarak yapan kullanıcılardır. Örneğin bir veznedar A hesabından B hesabına para transferi yapacaksa sadece bu transfer işlemini yapan programı kullanacaktır.

4-Uzmanlaşmış kullanıcılar : Geleneksel veri işlem çerçevesine uymayan veritabanı uygulamaları yazan kullanıcılardır. Bu uygulamalar arasında bilgisayar destekli dizayn sistemleri, bilgi temelli ve uzman sistemler, kompleks veritiplerini (grafik, ses gibi) saklayan sistemler ve çevre modelleme sistemleri sayılabilir.

1.11. Ayrıntılı Sistem Yapısı

Bir veritabanı sistemi herbiri sistemin tümünden sorumlu modellere ayrılmıştır. Veritabanı sisteminin fonksiyonlarından bazıları bilgisayarın işletim sisteminde sağlanır. Çoğu durumda işletim sistemi sadece temel servisleri sağladığından, veritabanı sistemi bu temel üzerine inşa edilmiştir. Dolayısıyla veritabanı sisteminin dizaynı, veritabanı ve işletim sistemi arasındaki arabirimin incelenmesini gerektirmektedir. Bir veritabanı sistemi aşağıdaki fonksiyonel elemanları ihtiva etmektedir.

1-Dosya yöneticisi : Görevi, disk depolama alanlarının⁴⁴ tahsisini ve disk üzerinde depolanan bilgileri göstermek için kullanılan veri yapısını yönetmektir.

2-Veritabanı yöneticisi : Görevi uygulama programları ve sisteme iletilen sorgularla, veritabanında depolanan, saklanan veriler arasında bir arabirim görevi yapmaktır.

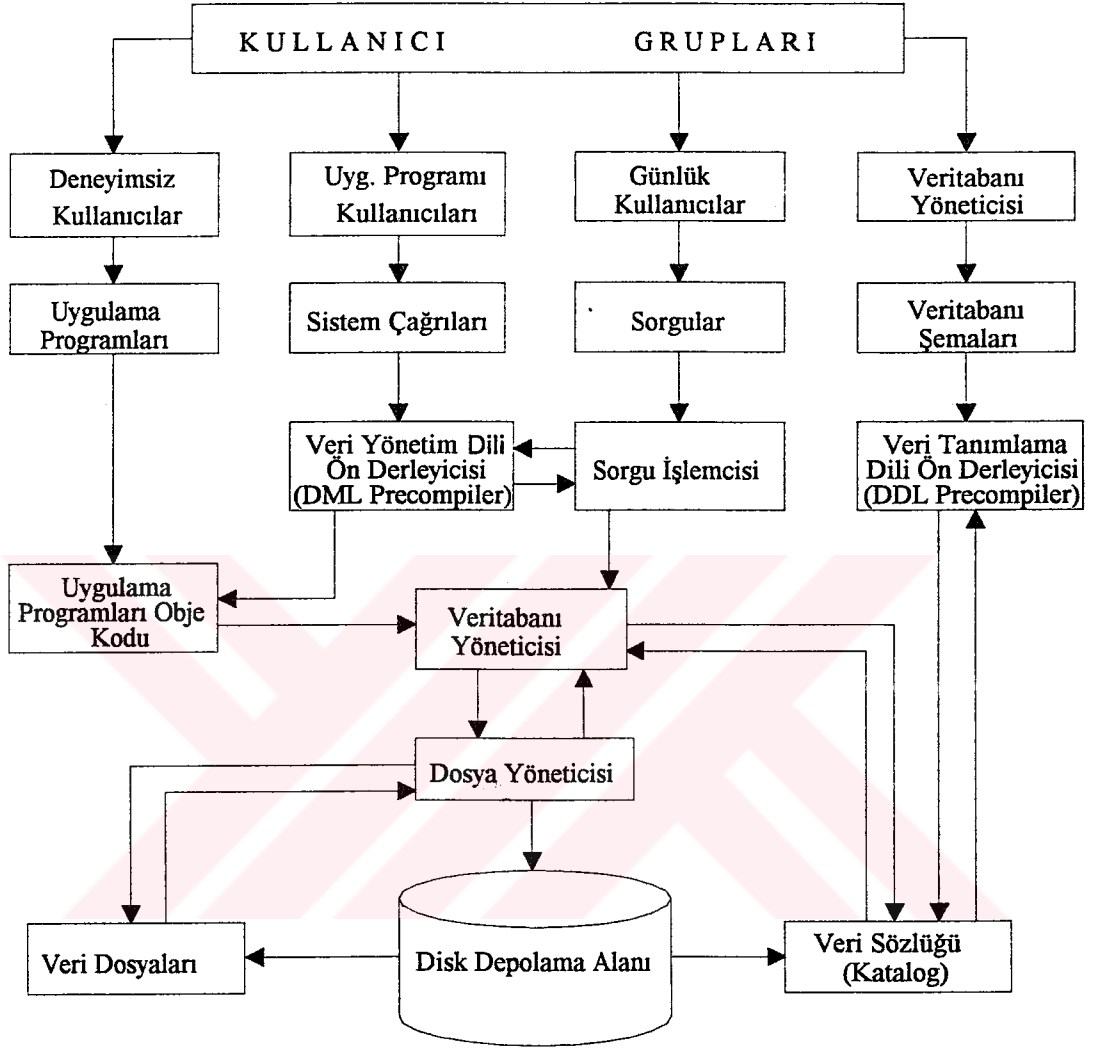
3-Sorgu işlemcisi : Bir sorgu diline ait ifadeleri veritabanı yöneticisinin anlayacağı şekilde düşük seviyeli komutlara çevirmek, ilave olarak kullanıcı isteklerini eşit fakat daha etkili formlara, yapılarla dönüştürmek suretiyle sorgunun çalışacağı en iyi stratejileri de tesbit etmek sorgu işlemcisinin görevidir.

4-DML önderleyicisi : Amacı, uygun kodlar üretebilmek için mutlaka sorgu işlemcisi ile etkileşimde bulunmak ve uygulama programındaki DML ifadelerini ana programın normal işlem çağrılarına çevirmektir.

5-DDL derleyicisi : Görevi, DDL ifadelerini metadata içeren tablo düzenlerine çevirmektir. Bu tablolar veri sözlüğünde saklanmaktadır.

⁴⁴ **Disk bellek/ disk deposu (disc storage) :** Manyetik yüzeylerinde verilerin saklandığı depolama cihazı.(Gazi Güder, 1986, Bilgi İşlem Terimleri Sözlüğü, Birsen Yayınevi, İstanbul, s. 141)

Aşağıdaki şekilde bu bileşenler ve aralarındaki bağlantılar gösterilmiştir⁴⁵.



Şekil 3. Ayrıntılı Sistem Yapısı

1.12. Varlık-İlişki Veri Modeli (Entity-Relationship Data Model)

Varlık-ilişki (E-R) veri modeli varlıklar (entities) olarak isimlendirilen temel nesneler ve bu nesneler arasındaki ilişkiler (relationship) topluluğundan meydana gelen gerçek bir dünya anlayışına dayanmaktadır. Veritabanları tasarlanırken öncelikle bu model düzenlenmektedir. Böyle bir şema, veritabanının tüm mantıksal yapısını gösterebilmektedir⁴⁶.

Varlık, diğer nesnelerden ayırdedilebilen ve var olan bir nesnedir. Varlık seti ise benzer türdeki varlıklar topluluğudur. Varlıkları birbirinden ayıran özellikler (attributes) varlığa ait nesneyi tanımlamaktadır. Varlık soyut -bir şahıs- veya somut -bir kavram- olabilir. Varlık

⁴⁵ Korth and Silberschatz, a.g.e., p. 15-16

⁴⁶ Korth and Silberschatz, a.g.e., p. 21

seti özellikler topluluğu ile gösterilmektedir. Örneğin bir bankada hesabı olan kişilerin tümü müşteri varlık seti; müşterinin adı, sicil no, cadde, şehir v.b. de müşteri varlık setinin özelliklerine karşılık gelmektedir. Benzer şekilde hesap varlık setinde hesap numarası ve bakiye, özellikleri temsil etmektedir. Böyle her bir özellik için alan (domain), denilen belli değerlerin girilebildiği sahalardır. Örneğin müşteri adı alanına sadece belli uzunlukta ve karakter türde, hesap numarası alanına da pozitif numerik değerlerin girilebildiği yerler vardır. Özellikler ise varlık setini bir alana ilişkilendiren bir fonksiyondur. Dolayısıyla bu modelde herbir varlık (alan, veri değeri) çifti ile gösterilmektedir. Örneğin :

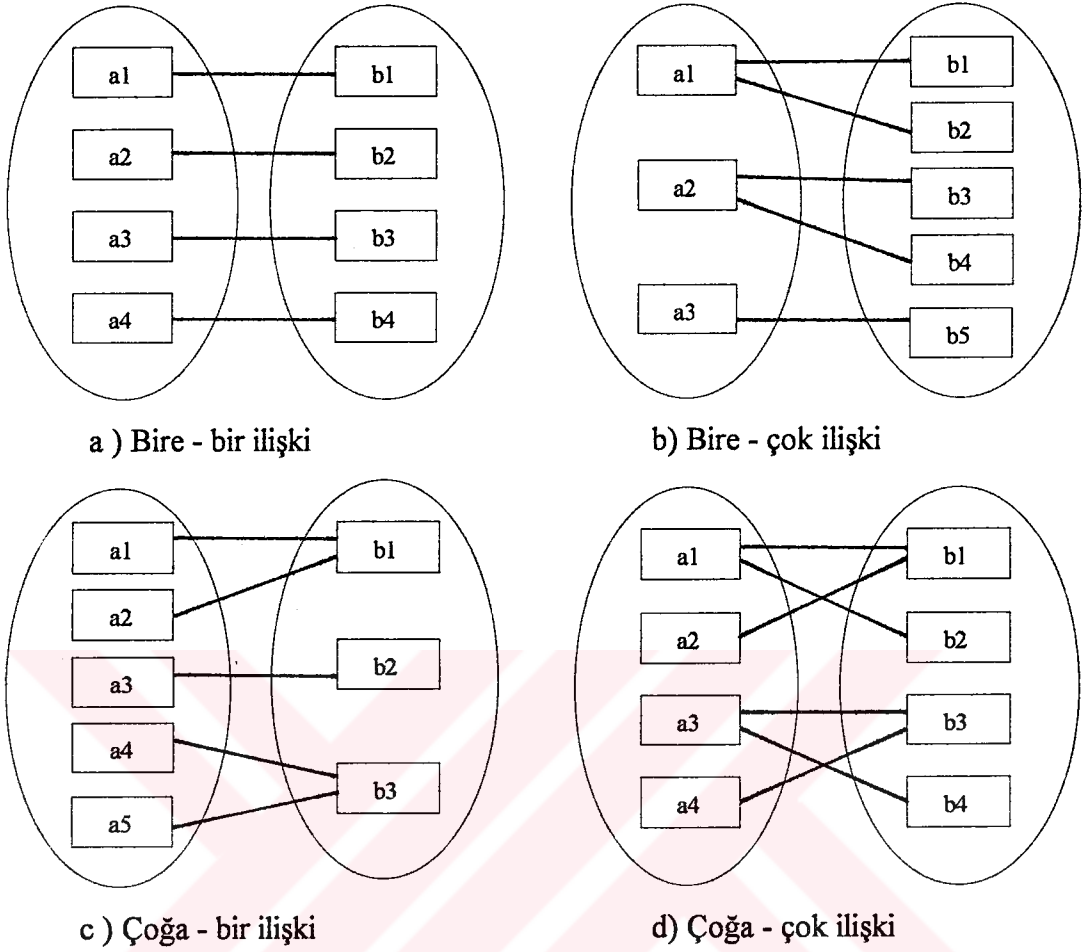
{(Adı, Kutlu), (Sicil_no, 890), (Cadde, Bayrak), (Şehir, Erzurum)} gibi.

Varlık seti kavramı programlama dillerindeki değişken tiplerinin tanımlanmasına benzemektedir. Tipi belirtilen bir değişken belli bir anda özel bir değere sahiptir. Yani değişken kavramı E-R modelindeki varlık kavramının yerini tutmaktadır. Zaten veritabanı da benzer özellikte ve değişik sayıda varlıkları ihtiva eden bir varlık seti koleksiyonudur.

Çeşitli varlıklar arasındaki bağlantılara ilişki, benzer türdeki ilişkiler topluluğuna da ilişki seti denmektedir. İlişki içerisinde varlığın üstlendiği görev onun rolü olarak bilinir. Roller ilişkilerin anlamlarını açıklamada çok kullanışlıdır. Bir E-R iş şeması, veritabanının mutlak uyması gereken kesin sınırlamaları belirtebilmektedir. Bu sınırlama şekillerinden birisi olan ilişkilerin derecelendirilmesi (mapping cardinalities), bir ilişki yoluyla varlıkların bir diğer varlığa nasıl bağlanabileceğini izah etmektedir. İlişki derecelendirme her ne kadar ikiden fazla varlık seti gerektiren ilişkilere katkıda bulursa da en çok ikili ilişki (binary relationship) modellerini tanımlarken kullanılmaktadır. A ve B varlık setleri arasındaki -ikili bir ilişki için- ilişki derecesi mutlaka aşağıdakilerden biri olmalıdır.

- 1-Bire-bir (one-to-one) : A veya B'deki bir varlık en fazla A veya B'deki bir varlıkla eşleşebilmektedir.
- 2-Bire-çok (one-to-many) : A'daki bir varlık B'deki varlıkların herhangi biri ile eşleşebilirken, B'deki varlıklar en fazla A'daki bir elemanla eşleşebilmektedir.
- 3-Çoğa-bir (many-to-one) : A'daki bir varlık en fazla B'deki bir elemanla eşleşirken, B'deki bir varlık A'daki herhangi bir elemanla eşleşebilmektedir.
- 4-Çoğa-çok (many-to-many) : A veya B'deki herhangi bir eleman B veya A'deki bir elemanla eşleşebilmektedir⁴⁷.

⁴⁷ Korth and Silberschatz, a.g.e., p. 22-28



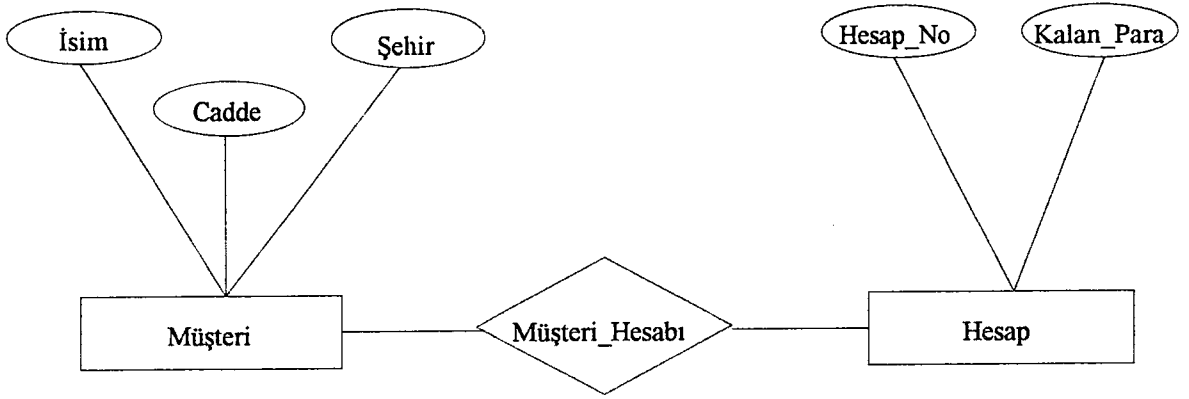
Şekil 4. İlişki dereceleri

Diğer bir sınırlandırma formu ise bağımlılık formudur (existence dependency). Varlık X'in mevcudiyeti varlık Y'nin var olmasına bağlıdır. Bunun anlamı örneğin Y varlığı silinirse X'in de silineceğidir. Y'ye baskın (dominant) X'e de ikincil (subordinate) varlık denir.

Hesap ve işlem varlık setleri arasında özel bir hesap ilişkisi olduğunu farzedelim. Bu ilişki hesaptan işleme doğru ve bire-bir ilişkidir. Her bir işlem hesap numarası ile birlikte ele alınmaktadır. Şayet hesap varlığı silinecek olursa onunla beraber ele alınmış olan tüm işlem varlıkları da silinmelidir. Ters bir durum sözkonusu değildir. Yani işlem varlıkları silinince veritabanındaki hesap varlığı etkilenmeyecektir. Dolayısıyla bu kayıt ilişkisindeki hesap varlık seti baskın, işlem varlık seti de bağımlı olmaktadır.

E-R diyagramlarında dikdörtgenler varlık setlerini, elipsler özellikleri, baklava biçimleri ilişki setlerini, çizgiler de özellikleri varlık setlerine (elipsleri dikdörtgenlere) ve varlık setlerini de ilişkilere (dikdörtgenleri baklava biçimlerine) bağlayan hatları -ki bu hatlara verilen isimlere rol denir- temsil etmektedirler⁴⁸.

⁴⁸ Korth and Silberschatz, a.g.e., p. 29-31



Şekil 5. Örnek bir E-R diyagramı

Veritabanı modeli oluşturulurken yapılması gereken önemli bir görev de varlıkların ve ilişkilerin nasıl ayırdedilebileceğini tanımlamaktır. Kavram olarak bireysel varlıklar ve ilişkiler farklılıklar göstermektedir. Fakat bir veritabanı perspektifinden bakılınca varlıkların özelliklerinde bulunan farklılıklar mutlaka izah edilmelidir. Bu tür farklılıkları oluşturmak için her bir varlık setine birincil anahtar (primary key-superkey) atanır.

Birincil anahtar varlık seti içerisindeki yegane bir varlığı ve ilişki setinde yer alan tek bir ilişkiyi toplu bir şekilde ele alarak tanıtıcı bir veya birden fazla özellikler (farklılıklar) topluluğudur. Örneğin hesap numarası özelliği müşteri varlık seti içindeki müşterileri ayırdedebilmek için yeterli olduğundan birincil anahtar olarak tanımlanabilir. Müşteri adı ise aynı isme sahip birden fazla kişi olabileceğinden dolayı birincil anahtar olamaz. Varlık setindeki özellikler birincil anahtar formu elde etmek için yeterli olmayabilir. Bu tür varlık seti zayıf (weak), birincil anahtara sahip varlık seti ise güçlü (strong) bir varlık setidir.

Veritabanının tüm mantıksal yapısı E-R diyagramlarıyla grafiksel olarak izah edilebilmektedir. E-R diyagramına uydurulan bir veritabanı, tablolar topluluğu tarafından da gösterilebilmektedir. E-R modelinin bir sakıncası ilişkilerarası ilişkileri izah edememesidir. Çözüm olarak ilişkiler setinin yüksek seviyeli bir varlık seti gibi davrandığı ve diğer bir soyutlama şekli olan biraraya getirme (aggregation) metodu önerilmektedir⁴⁹.

1.13. İlişkisel Model (Relational Model)

Günümüze kadar pek çok farklı dosya tipi geliştirilmiş ve kullanılmıştır. Fakat zaman içinde dosyalar daha karmaşıklaşmış ve programcılarının işleri zorlaşmıştır. 1970'lerin başında ise farklı bir dosya modeli ortaya çıkmıştır. İlişkisel dosya adı verilen bu yapı oldukça basittir ve bütün bilgiler tablo benzeri bir dosyada saklanmaktadır. Bu tabloda satırlara kayıt, kolonlara da alan denilmektedir. Her dosyanın bir kayıt deseni bulunmaktadır ve bu desen dosyadaki bütün kayıtlar için geçerlidir. Yani her kayıt aynı sayıda ve türde alan içermektedir. Kaydın alanlarına bakılarak tekrar olup olmadığı

⁴⁹ Korth and Silberschatz, a.g.e., p. 27-41

kontrol edilebilmektedir. Bu alan veya alanlar grubuna anahtar ismi verilmektedir. Her ilişkisel dosyanın bir anahtarı vardır. Anahtar içindeki alanların minimum seviyede olmasına çalışılır. Genellikle anahtar üzerinden söz konusu dosya için bir indeks vardır. Her yeni kayıt eklemede bu indeks kullanılarak arama yapılır. Eklenilmesi düşünülen yeni kaydın anahtar alanlarındaki bilginin bir benzeri dosyada mevcut ise bu yeni kaydın eklenilmesi reddedilir. DBMS'lerde anahtar alanların güncelleştirilmesine ve boş bırakılmasına izin verilmez.

Veritabanı konusundaki son araştırmalar "ilişkisel sistemler sonrası" olarak adlandırılabilir bir eğilime sahiptirler. Bu araştırmaların bir kısmı ilişkisel sistemlere tamamıyla bağımlı kalarak onların üzerinde çalışmaya giderken, bazıları ise tamamıyla farklı sistemler oluşturmaya çalışmışlardır. Bu yeni araştırma sahalarına örnek olarak Deductive, Semantic, Universal Relation, Object Oriented, Extendible ve Expert sistemler verilebilir. Fakat bunlar aynı zamanda birer araştırma kolu oldukları için bu kategorilerde pek az ürün bulunmaktadır. Kabul edilmesi gereken bir gerçek var ki o da ilişkisel sistemlerin güncel veritabanı pazarında baskın bir eğilimi temsil ettiği ve veritabanı tarihinde önemli bir gelişme olduğu gerçeğidir⁵⁰.

Tarihi bir perspektiften bakılınca ilişkisel veri modeli nisbeten yenidir. İlk veritabanı sistemleri ya hiyerarşik ya da şebeke modellerine dayandırılmıştı. İlişkisel veritabanlarının yapısı büyük bilgisayar sistemleri için geliştirilmiştir. Veri elemanları arasındaki ilişkileri tanımlayan basit yapısı nedeniyle son zamanlarda çok uygulanan bir model olmuştur. Bu yapı ile programcı olmayanlar dahi büyük miktardaki bilgileri tanzim edebilmekte ve etkileşimli olarak işleyebilmektedirler⁵¹.

Bir sistemin ilişkisel olduğunu söyleyebilmek için aşağıdaki nitelikleri taşıması gerekmektedir.

- 1- Verilerin tablolarda depolanması
- 2- Kullanıcıya açık göstergeler veya bağlantıların olmaması
- 3- Sorgu dilinin tamamen ilişkisel olması
- 4- Sorgulamada tekrarların (eş satırların) bulunmaması

İlişkisel veritabanı, kullanıcılar tarafından bir tablolar (ilişkiler) kümesi olarak algılanmaktadır. "Kullanıcı tarafından algılanan" ifadesi çok önemlidir. Kullanıcıyı sistemin donanım ve işletim sistemine bağlı alt düzey detaylardan yalıtmakta ve kullanıcının veri üzerindeki işlemlerini basit tablo işlemlerine dönüştürmektedir.

⁵⁰ Anonymous, Temmuz 1993, Bilgisayar (Datamation) Dergisi, YA-SA Yayıncılık, Yıl :16, Sayı : 147, s. 38

⁵¹ Korth and Silberschatz, a.g.e., p. 45

İlişkisel sistemleri, ilişkisel olmayan sistemlerden ayırt eden en önemli farklılıklardan birisi, ilişkisel sistemlerde kullanıcıların verileri sadece tablolar halinde görmesi ve kullanması mümkün olduğu halde, ilişkisel olmayan bir sistemde, diğer veri yapılarını da görmesi ve kullanabilmesidir. Diğer veri yapıları beraberinde daha farklı ve karmaşık işlemleri de gerektirebilmektedir. Örneğin hiyerarşik bir sistem olan IMS'de veriler kullanıcıya ağaç yapılarından oluşmuş kümeler şeklinde sunulmakta ve bu yapıları manipüle etmek veya ağaçları yukarı-aşağı taramak için hazırlanmış işlemler bulunmaktadır⁵².

İlişkisel olmayan sistemlerin dezavantajı ise veritabanının tasarımı esnasında gözden kaçan hataların, uygulamada fark edilen hatalar sebebiyle yapılacak değişikliklerin veriler üzerinde olumsuz bir etkiye sebep olmasıdır. Temel veri yapısı olarak tablo kullanan, ancak izin verilebilir bağlantılara sınırlamalar getiren sistemler ilişkisel olarak görülmez. Bu tür sistemler için düz (tabular) terimi kullanılmaktadır. İlişkisel model veri elemanlarını sıralar (rows) ve sütünlardan (columns) ibaret olan iki boyutlu bir tablo içinde organize etmektedir. Her sıra veritabanı içinde bir girişe ait olan bilgileri içermektedir. Bir sıra içindeki veri de, her biri tablo içinde bir kolonu işgal eden birçok kaleme bölünmektedir. Tabloların özellikleri arasında eş satırların olmaması, kolon ve satır sıralarının önemsiz olması, her bir satır ve sütun kesişiminde sadece bir değer bulunması sayılabilir.

Tablo 2. Örnek bir ilişkisel veritabanı tablosu

Oğrenci adı	Ders	Görevli
Ahmet	İş Yönetimi 101	Veli Akçam
Selami	İş Yönetimi 101	Veli Akçam
Mahmut	Ekonomi 100	Veli Akçam
Selami	Ekonomi 100	Veli Akçam
İhsan	Ekonomi 100	Veli Akçam
Ahmet	Ekonomi 100	Veli Akçam
Nesrin	Tarih 102	Mustafa Çevik
Fuat	Tarih 102	Mustafa Çevik
Ahmet	Tarih 102	Mustafa Çevik

Yukarıdaki tabloda görüldüğü gibi veritabanı içindeki elemanlar 9 satırlı⁵³ ve 3 kolonlu bir cetvel halinde organize edilmiştir. Her sıra bir veri kaydı (data record) ve her kolon bir veri alanı (data field) olarak isimlendirilir. Her veri kaydı girildikçe sıra numarasına benzer bir numara verilmekte ve bu numara veritabanı içinde depolanan veri kayıtlarının sırasını temsil etmektedir. Kaydın farklı parçaları farklı kolonlar veya alanlar içinde depolanmakta, sonuç olarak veritabanı içindeki herhangi bir veri elemanı onun kayıt numarası ve alan ismi ile tanımlanmaktadır.

⁵² Anonymous, Bilgisayar (Datamation), a.g.d., s. 38

⁵³ Tablolarda yer alan satırlara "tuple" denilmektedir.

Bir ilişkisel veritabanı kütüğü iki ana parçadan; veri kayıtlarının yapısından ve verinin kendisinden oluşmaktadır.

1-Veri kayıtları : Bir veri kaydı, veri kalemlerinin tek bir girişini tutmaktadır. Bu kayıtlar veritabanı içinde genellikle girildikleri sırada düzenlenirler. Örneğin, sınıf veritabanı içinde öğrenci_adı, ders ve görevli alanları bir veri kaydını oluşturmaktadır.

2-Veri alanları : Veri kaydı içinde tek bir veri parçasını tutan depolama birimidir. Her veri alanına veritabanı içinde tanımlanan bir isim verilmektedir. Alan ismi harfler, numaralar ve bazı sembollerin birleşiminden meydana gelen sabit veya değişken uzunluklu karakterler olabilmektedir. Veri alanının muhtevası alfasayısal veya sayısal bir değer içerebilmektedir. İlave olarak tarih, zaman, mantıksal veri alanları da tanımlanabilmektedir. Veritabanının yapısı, veri kaydı içindeki her alanın ayrıntılı tariflerini kapsar. Bu ayrıntılar şunlardır.

a- Alan adı (field name) : Veri alanının adı veya tanıtımı

b- Alan tipi (field type) : Veri alanının cinsi

c- Alan genişliği (field width) : Veri alanının boyutu

Bu yapıların tanımlanması birkaç amaca hizmet etmektedir. Veri işlenirken yapının içinde belirtilen alan isimleri, bu alanlar içinde depolanan veriyi yeniden çağırmak ve başvurmak için kullanılmaktadır. Örneğin, öğrenci_adı alanı bir veritabanı uygulamasının başından sonuna kadar belli bir öğrenci adı ile bağlantılı, bütün bilgilerin yerini bulmak için kullanılabilir. Kullanılacak veri tipinin tanımlanması, bilginin nasıl kullanılacağını dikte etmektedir.

Bir veritabanı üzerinde yapılabilecek işlemler aşağıda verilmiştir.

a- Veritabanı içeriğinin muhafazası ve güncelleştirilmesi,

b- Belirtilen bir tanımlamalar setini karşılayan verinin yerinin bulunması ve bilginin açıklanması,

c- Bir veri grubunun, önceden tayin edilmiş bir düzen içinde sıralanması veya yeniden düzenlenmesi,

d- Farklı veritabanı kütükleri içinde veri ögelerinin bu ögeler üzerinden dolaylı olarak indekslenmesi ve bağlantılarının yapılması.

Veri bakım işlemi, verinin veritabanına ilavesi (insert), içindekilerin bir kısmını veya tamamını değiştirme (update) ve ondan ögeleri silme (delete) gibi gerekli modifikasyon işlemleridir. Veritabanı yönetiminde diğer önemli bir işlev olan ögenin kayıt numarasına ve alan ismine bakmak suretiyle veritabanı içindeki verinin yerinin bulunması ve çıkarılıp alınması (query) ise, ilişkisel bir veritabanı için çok kolay bir iştir. Veritabanındaki veri kayıtlarını sıralama (sort), yeniden düzenleme, değişik veritabanı kütüklerindeki veri elemanlarını birleştirme (join) ilişkisel veritabanılarının diğer güçlü özellikleridir. İlişkiler ve bağlar göstergelerle değil veri üzerinden kurulmaktadır. Kullanıcı veriye ulaşım yolunu tanımlamak ve verinin fiziksel yerleşimini bilmek zorunda değildir. İlişkisel sorgulama dillerinde kullanılan cebir, ilişkisel cebir olup temel olarak beş operasyona sahiptir. Seçme

(select), tasarım (project), kartezyen sonuç (cartesian-product), birleşim (union) ve fark (difference). Tüm bu operasyonlar yeni bir ilişki meydana getirmektedirler. İlişkisel cebir dilleri veritabanlarının sıradan kullanıcıları için uygun olmayan kısa ve resmi diller olduklarından, ticari veritabanı sistemleri yapısal ve yapısal olmayan yaklaşımlara ait sözdizimlerini (syntax) daha çok kullanmaktadırlar. Ticari dillerin en önde gelenleri SQL, QUEL ve QBE (Query By Example) olarak bilinmektedir. Veritabanı sistem ürünleri daha çok kullanıcıya yönelik (user-friendly) sorgu dilleri gerektirdiğinden, bunlardan QUEL satır ilişkili hesaplara, QBE alan ilişkili hesaplara dayandırılmıştır. SQL ise ilişkisel cebirin ve hesaplama yapılarının kombinasyonunu kullanmaktadır. Bu üç dil sadece veritabanı araştırma sistemlerinde değil ticari alanda da tesirli olmaktadır. Bu dillere her ne kadar "sorgu dilleri" deniyorsa da bu yanlış bir kullanım olmaktadır. SQL, QUEL ve QBE bir veritabanı sorgulamanın ötesinde kabiliyetlere -veri yapısını tanımlama, veriyi biçimlendirme, güvenlik sınırlarını tanımlama gibi- sahiptirler.

1.13.1. Görüntüler (views)

Kullanıcıların araya bilgi girme, silme, sorgulama, birleştirme işlemlerini yapabilmeleri için kavramsal modelin tümünü görmeleri gerekmemektedir. Güvenlik tedbirlerinden dolayı bazı verilerin bazı kullanıcılardan gizlenmesi gerekmektedir. Görüntü terimi ile kavramsal modelin bir bölümü olmayıp da kullanıcı tarafından gerçek bir ilişki gibi görünen herhangi bir ilişki anlatılmaktadır. Kavramsal modelde gerçek ilişkiler insert, update, delete operasyonları ile her an değiştirilebileceği için genellikle görüntüleri saklamak pek mümkün olmamaktadır. Bunun yerine bu görüntüyü elde edecek her bir sorgulama için yeniden hesaplanır ve oluşturulur. Özetle görüntüler veritabanı sorgulamalarını basitleştirmede kullanılan bir mekanizmadır⁵⁴.

1.13.2. 4. Normal Form

Veritabanlarında kullanılan dosyaların 4. Normal Form'da olması pek çok problemi engellemektedir. 4. Normal Form, kısaca ilişkisel tipli dosyalarda anahtar olmayan alanların sadece ve sadece anahtar alan grubuna bağımlı olması ve başka hiçbir bağımlılığın olmaması özelliğidir⁵⁵. Aşağıda istenmeyen bir takım bağımsızlıkların olduğu bir dosya örnek olarak alınacak ve değişik işlemlerde (güncelleme, iptal ve ekleme) oluşabilecek veri bozulmaları incelenecektir.

Tablo 3. Örnek depo dosyası

Depo no	Personel no	İsim	Maas	Şehir	İlceim
102	1001	Ali	47.000.000	İzmir	15000
101	1003	Canan	42.000.000	İstanbul	13500
102	1004	Hakan	38.000.000	İzmir	15000
103	1006	Yeşim	45.000.000	Ankara	13500
101	1007	Yavuz	35.000.000	İstanbul	13500

⁵⁴ Korth and Silberschatz, a.g.e., p. 50-99

⁵⁵ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 207-208

Yukarıdaki tabloda yer alan depo dosyasında isim, maas, depo_no personel_no'ya; şehir ve hacim ise depo_no'ya bağımlıdır. Bu dosyanın anahtarı personel_no'dur. Anahtar olmayan alanlar ise, anahtar alan olan personel_no'ya bağımlı değildir ve bir takım istenmeyen bağımlılıklar mevcuttur.

a-Güncelleştirme : 102 nolu deponun İzmir'den Bursa'ya taşındığını düşünelim. Bu durumda dosyada düzeltme yapmak gerekecektir. Sadece 1001 nolu personelin depo şehrinin değiştirilmesi yeterli olmayacak bütün 102 nolu depoda çalışanların şehirlerini düzeltmek icap edecektir. Sadece 1001 nolu personelin şehri Bursa olarak düzeltilirse veritabanında bir tutarsızlık olacaktır. Şöyleki, 1001 nolu personel kaydına göre 102 nolu depo Bursa'dadır, halbuki 1004 nolu personel kaydına göre ise 102 nolu depo İzmir'dedir.

b-İptal : Bu dosyadaki 1006 nolu personel iptal edilecek olursa 103 nolu deponun Ankara'da olduğu bilgisi yok olacaktır. Böyle beklenmedik bir şekilde yok edilen fakat farkedilemeyen bilgiler başka işlemlerin yan etkileri sonucu uçup gidecektir.

c-Kayıt ilave : Eğer yeni bir depo açılırsa bu depoyu dosyaya yeni bir kayıt olarak eklemek problemler oluşturacaktır. Ekleme yapıldığında bu kayıta sadece depo_no, şehir ve hacim bilgileri var olacaktır. Halbuki bu dosyanın anahtarı Depo_no ve Personel_no'dan oluşmaktadır. Anahtarın bir parçası olan Personel_no boş kalacaktır ki bu duruma veritabanlarında izin verilmez.

Bahsedilen problemler ile karşılaşmamak için bu tür dosyaları iki veya daha çok alt kümelere ayırmak gerekecektir. Oluşan yeni dosyalar 4. Normal Form'da olmak zorundadır. Oluşan bu yeni dosyalar istenilen formda değilse alt kümelere ayırma işlemi devam edecektir. Örnek olarak Tablo 10.'da verilen Depo dosyası, Depo ve Personel dosyası olarak iki kısma ayrılarak 4. Normal Form'a dönüştürülmekte, problemler aşağıdaki gibi giderilmektedir.

a-Güncelleştirme : Eğer 102 nolu depo İzmir'den Bursa'ya taşınırsa bu düzeltme sadece Depo dosyasında bir kayıta yapılacak başka bir işleme gerek olmayacaktır. Tutarsızlıklar ortaya çıkmayacaktır.

b-İptal : Personel dosyasındaki 1006 nolu personelin kaydı iptal edilecek olursa hiçbir şekilde 103 nolu deponun Ankara'da olduğu bilgisi kaybolmayacaktır. Çünkü bu bilgi farklı bir dosyada kayıtlıdır.

c-Kayıt ekleme : Yeni bir depo servise açılacak olursa bununla ilgili kayıt hiçbir engel olmaksızın Depo dosyasına eklenebilecektir⁵⁶.

1.14. Şebeke Modeli (Network Model)

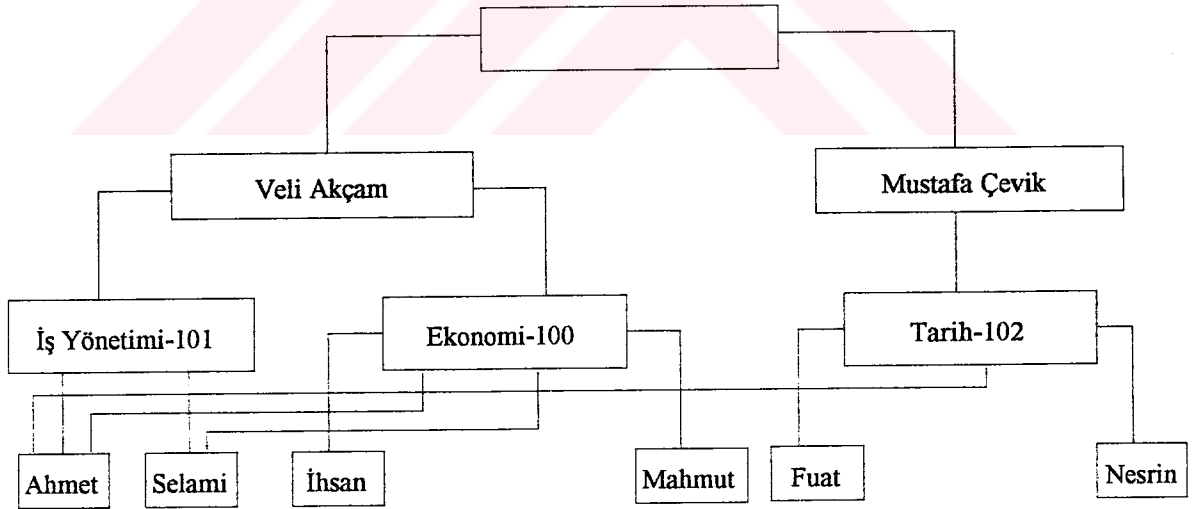
İlişkisel modelde veri ve veriler arasındaki ilişkiler bir tablo topluluğu tarafından gösterilirken, şebeke modelinde veri kayıtlar topluluğu tarafından ve veriler arası ilişkiler de hatlar (links) tarafından gösterilmektedir. Kayıt bir çok yönden varlık-ilişki modelindeki varlık kavramına benzemektedir. Her bir kayıt, herbiri sadece bir veri değeri içeren sahalar (fields) veya özellikler topluluğudur. Hat ise iki kayıt arasındaki kesin bir ilişkiyi temsil etmektedir⁵⁷. 1960'ların sonunda şebeke modeline dayalı bir çok ticari

⁵⁶ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 208-210

⁵⁷ Korth and Silberschatz, a.g.e., p. 107

veritabanı sistemi ortaya çıkmıştır. Bu sistemler Database Task Group (DBTG) tarafından CODASYL⁵⁸ grubu içinde geniş bir şekilde incelenmiştir. DBTG modelinde sadece bire-bir ve bire-çok ilişkiler kullanılmıştır. Uygulamayı basitleştirmek için çoğa-çok ilişkilere müsaade edilmemektedir. Birbirine bağlı iki kayıt tipinden ibaret olan bir veri yapı diyagramı, DBTG modelinde DBTG seti olarak belirtilir. DBTG setlerinin herbirinde setin "sahibi" ve "üyesi" olarak dizayn edilmiş bir kayıt tipi vardır. Bu setler herhangi bir sayıda olabilir.

DBTG modelinin veri yönetim dili, ana dilde saklı olan bir çok komuttan meydana gelmektedir. Bu emirler yerel olarak tanımlanmış değişkenlere olduğu kadar veritabanı değişkenlerine de -bir program çalışma alanı oluşturarak- erişip yönetmektedirler. En sık "find" ve "get" komutları kullanılmaktadır. Verilerin güncelleştirilmesinde de oldukça farklı komutlar kullanılmakta, mevcut kayıtlar üzerindeki değişiklikler "modify" komutu ile yapılmaktadır. "Store" ve "erase" işlemleri yoluyla yeni kayıtların oluşturulup, silinmesi sağlanabilmektedir. Belirli bir küme oluşumuna kayıt ilave edip ondan bilgi almak için de "connect, disconnect, reconnect" komutları kullanılmaktadır. Yeni bir set tanımladığında ise, bir set oluşumundan bir başkasına üye kayıtlarının nasıl aktarılabilceği veya çıkarılabilceği açıkça belirtilmelidir. Daha önce eklenmiş olan bir üye kaydının set oluşumundan nasıl ve ne zaman çıkarılabilceği konusunda ise sınırlamalar mevcuttur. Bu sınırlamalar set tanımı esnasında çıkarma işleminin sabit, mecburi ve seçimlik olma durumları belirtilerek yapılmaktadır⁵⁹. Aşağıda bir şebeke tipli veritabanının yapısı yer almaktadır.



Şekil 6. Şebeke Ağaç Modeli

⁵⁸ CODASYL : Conference On DATA SYstem Languages sözcüklerinin başharflerinden oluşan bir kısaltmadır. COBOL programlama dili bu konferansta geliştirilmiştir. (Güder, a.g.e., s. 79)

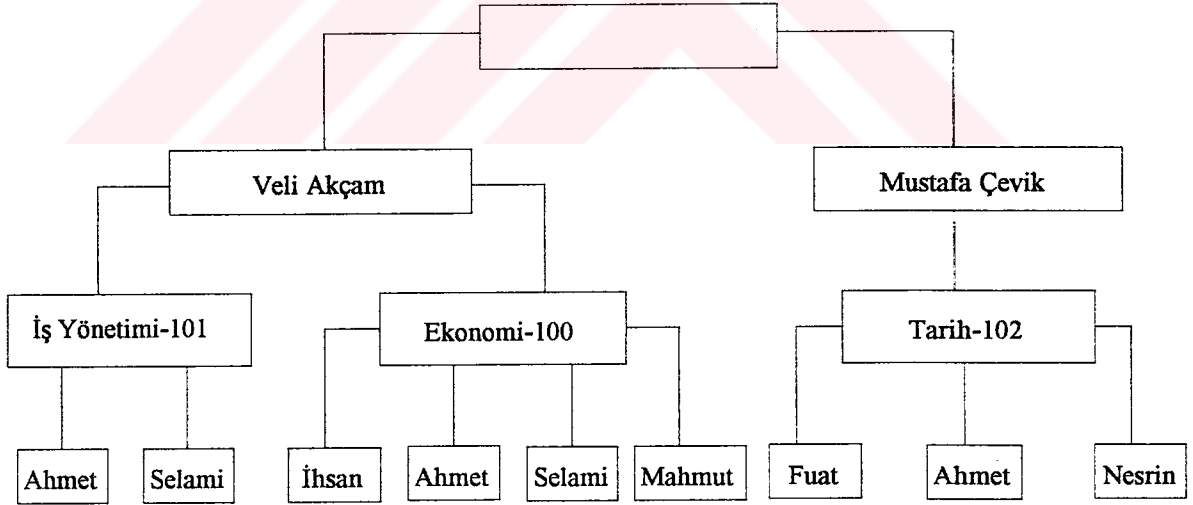
⁵⁹ Korth and Silberschatz, a.g.e., p. 115-136

1.15. Hiyerarşik Model (Hierarchical Model)

Hiyerarşik bir veritabanı birbirlerine hatlar vasıtasıyla bağlanmış olan kayıtlar topluluğundan ibarettir. Hiyerarşik modelde veriler ve aralarındaki ilişkiler kayıtlar ve hatlar tarafından temsil edilmeleri açısından şebeke sistemine benzemektedir. Fakat ondan farklıdır. Çünkü kayıtlar ağaç yapılar topluluğu ile organize edilmektedir. Ağaç yapı şeması, hiyerarşik veritabanı için bir plandır. Bu ağacın kökü (root) bir ölü nokta (dummy node) olup bu düğümün elemanları çeşitli kayıt tiplerini içermektedir. Veritabanı bu tür ağaçlardan oluşan bir ormanı andırmaktadır. Bilginin hangi sıklıkta ve ne şekilde kullanılacağı önceden belirlenmekte ve bu bilgilere sıralı bir şekilde erişilmektedir. IMS ve DL/1 bu modelin örneklerindendir.

Veri yönetim dili ana dilde mevcut olan birtakım emirlerden oluşmaktadır. Bu emirler yerel ve veritabanı değişkenlerine erişebilmekte, onları kullanabilmekte, bu işlemleri de program çalışma alanında yürütebilmektedir. Veritabanındaki bilgilere erişim, mevcut göstergeyi (pointer) bulunan kayda yerleştiren ve daha sonra bu emri uygun program çalışma alanına kopyalayan “get”, ekleme ve silme işlemleri de “replace” komutu ile yapılmaktadır.

Çoğa-çok ilişkiler durumunda veritabanı ağaç yapısının organizasyonu korunmak isteniyorsa, kayıtlar tekrarlanmalıdır. Kayıt tekrarlarının iki önemli sakıncası vardır. Birincisi, güncelleştirme yapılırken veri tutarsızlığı ortaya çıkmakta ayrıca yer israfı kaçınılmaz olmaktadır⁶⁰. Aşağıda hiyerarşik tipli bir veritabanının yapısı yer almaktadır.



Şekil 7. Hiyerarşik Ağaç Modeli

⁶⁰ Korth and Silberschatz, a.g.e., p. 143-166

1.16. Dosya ve Sistem Yapısı

Kavramsal ve mantıksal seviyede veritabanı ilişkisel modelde tablolar, şebeke modelinde kayıt tipi, hiyerarşik modelde ise ağaç yapılar topluluğu olarak görüldüğünü, veritabanının mantıksal modelinin, veritabanı kullanıcıları için en doğru seviyedeki faaliyet merkezi olduğunu, veritabanı için asıl amaç veriye yapılacak erişimi kolay ve basit kılması olduğunu, kullanıcıların fiziki detaylardan haberlerinin olmasının gerekmediğini önceki bölümlerde görmüştük. Mamafih, kullanıcıları memnun eden veya etmeyen önemli bir faktör veritabanı performansıdır. Yapılan bir isteğe alınan cevap süresinin uzun olmasına bir dezavantaj olarak bakılmaktadır. Bir sistemin performansı veriyi göstermek için kullanılan veri yapılarının etkinliğine ve bu veri yapıları üzerinde icra edilecek operasyonların ne etkinlikte olacağına bağlıdır. Çoğu zaman işletim sistemi, temel servisleri (BIOS-Basic Input Output System) sağlamakta ve veritabanı bu temele oturtulmaktadır. Bir veritabanı sistemi aşağıdaki fonksiyonel bileşenlerden oluşmaktadır⁶¹.

- 1- Disk üzerinde depolanan bilgiyi göstermede kullanılan veri yapılarını ve disk bellek alanının tahsisini yöneten dosya yöneticisi (file manager)
- 2- Disk bellek ve ana hafıza arasındaki bilgi transferini yürüten arabellek yöneticisi (buffer manager)
- 3- Sorgu dili ifadelerini düşük-seviyeli bir dile çevirmekle yükümlü olan sorgu ayrıştırıcı (query parser)
- 4- Kullanıcı isteğini, eşit fakat daha etkili bir forma sokarak sorgunun en etkin şekilde çalışabileceği strateji tesbitini yapmakla vazifeli olan strateji seçici (strategy selector)
- 5- Bütünlük sınırlarını ve veriye erişecek kullanıcı yetkilerini test eden yetki ve bütünlük yöneticisi (authorization and integrity manager)
- 6- Sistem arızalarına karşı veritabanı tutarlılığının doğru olarak sürmesini sağlayan iyileştirme/onarma yöneticisi (recovery manager)
- 7- Eşzamanlı işlemlerin birbirini etkilemeden veritabanınca cevaplanmasını sağlayan eşzamanlı⁶² idareci (concurrency controller).

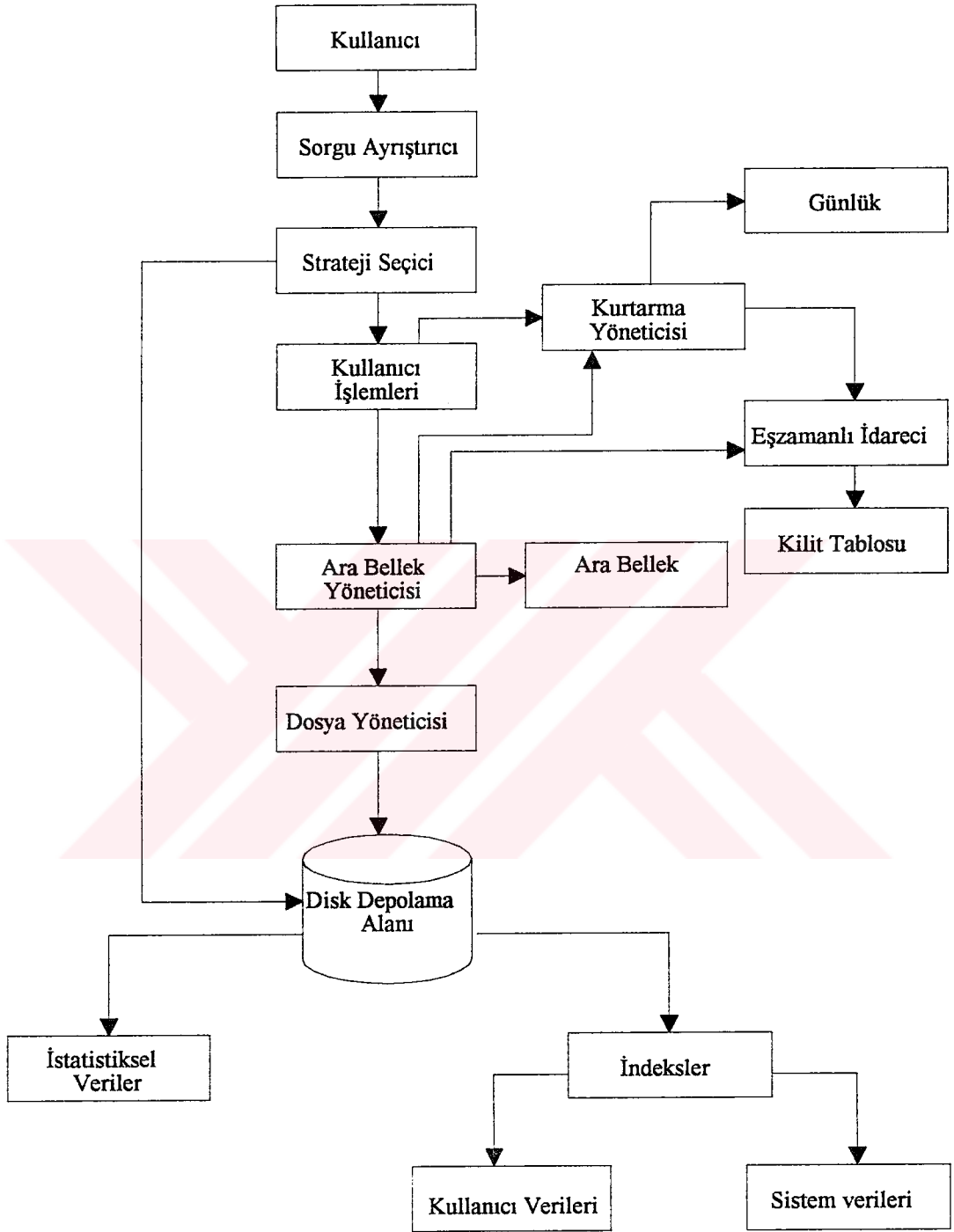
Fiziksel bir sistemin gerektirdiği veri yapıları ise şunlardır.

- 1-Veri dosyaları (data files) : Veritabanının kendisinin saklandığı yerdir.
- 2-Sistem veri dosyaları (system data files) : Veritabanı yapısı hakkındaki bilgilerin tutulduğu bir yer olup oldukça yoğun bir kullanıma sahiptir.
- 3-İndeksler (indices) : Özel değerlerin tutulduğu veri parçalarına daha hızlı erişimi temin eden mekanizmalardır.
- 4-İstatistiksel veriler (statistical data) : Veritabanında veri ile alakalı bilgileri tutar. Bu bilgiler strateji seçici tarafından kullanılır.

⁶¹ Korth and Silberschatz, a.g.e., p. 229

⁶² **Eşzamanlı (concurrency)** : Elektronik bilgi işlemde farklı yerlerde çok sayıda komutun veya işlemin aynı anda uygulanması ile ilgili çeşitli yöntemleri belirtir. Daha ziyade çok işlemcili (multi processor) sistemlerle ilgilidir.(Güder, a.g.e., s. 91)

Bu sistemin yapısı aşağıda gösterilmiştir⁶³.



Şekil 8. Sistem Yapısı

⁶³ Korth and Silberschatz, a.g.e., p. 230-231

1.17. İndeksler

Bir dosya için indeks, aynen kütüphane sistemlerindeki bir katalog gibi çalışmaktadır. Kütüphaneden herhangi bir yazara ait bir yayın aranıyorsa yazar kataloguna bakılır. Yazar katalogunda bulunan kartlar, kitabın nerede bulunabileceğini söyleyebilir. Kataloglarda aramayı kolaylaştırmak için kartlar alfabetik sırada saklanmaktadır. Dolayısıyla aradığınızı bulmak için her kartı kontrol etmek mecburiyeti yoktur.

Sorguların çoğunda, bir dosyadaki kayıtların çok az bir bölümüne müracaat edilmektedir. Bu kayıtları ararken geçen zaman süresini en aza indirmek için veritabanında indeksler oluşturulur. Gerçek dünyadaki veritabanları yukarıda tanımlanan yapıları indeksler tarafından ele alınamayacak bir şekilde büyük olabilmektedirler. Bunun yerine daha gelişmiş indeksleme teknikleri kullanılmaktadır. En iyi teknik, veritabanı uygulama yapısına uyan tekniktir. Her bir teknik ayrıca şu açılardan da değerlendirilmelidir.

1-Erişim zamanı (access time) : Sorguda kullanılan tekniğin özel bir veri parçası değerini bulma zamanıdır.

2-Araya ekleme zamanı (insertion time) : Yeni bir veri parçası ekleme zamanı. Bu zaman dilimi araya eklenecek yeni veri parçasının doğru yerini tayin etmeyi ve indeks yapısını güncelleştirilmesini de kapsar.

3-Silme zamanı (deletion time) : Veri elemanını silmek için geçen süredir. Silinecek kaydın bulunması ve indeks yapısının güncelleştirilmesi bu zaman dilimini teşkil eder.

4-Boşluk kullanımı (space overhead) : İndeks yapısı tarafından esneklik sağlamak ve artan bir performansa ulaşmak için tutulan ilave boşlukların kullanılabilmesidir⁶⁴.

Bir dosya için sık sık bir veya daha fazla indeks fonksiyonuna ihtiyaç duyulmaktadır. Kütüphane örneğimize dönecek olursak çoğu kütüphaneler çeşitli katalog kart sistemi sunmaktadırlar. Yazar, konu, başlık v.s. bir dosyadaki kayıtları aramak için kullanılan özellik veya özellikler topluluğu arama anahtarı (search key) olarak adlandırılır. Tanımlanan bu anahtar birincil (primary), aday (candidate) ve süper anahtardan farklıdır. Anahtar için bu çift anlamın uygulamada iyice tayin edilmesi gerekmektedir. Sıralı indeks dosyaları⁶⁵ veritabanı sistemlerinde kullanılan çok eski indeks şemalarından biridir. İsminden de anlaşıldığı üzere bir sıralı dosya ve bir indeksten oluşmaktadır. Bu dosyalar hem bir dosyanın sıralı olarak işlenmesini hemde özel kayıtlı rastgele erişimi (random access) gerektiren uygulamalarda kullanılmaktadır.

Sıralı bir dosya, kayıtların etkin şekilde işlenmesi için, temeli arama anahtarına dayanacak şekilde sıralı olarak oluşturulmalıdır. Kayıtların, arama anahtarı düzeninde hızlı bir biçimde yeniden ele alınabilmesi için birbirlerine göstergelerle bağlıdır. Göstergeler, arama düzenindeki bir sonraki kayda işaret etmektedir. Bu tür dosya işlemlerinde erişim bloklarının sayısını azaltmak için kayıtlar ya fiziksel olarak arama düzeninde veya

⁶⁴ Korth and Silberschatz, a.g.e., p. 265-266

⁶⁵ Sıralı indeks dosyaları (index sequential files) : Kayıtların sıralı olarak organize edildiği indekslerle referanslandırılan veri organizasyon yöntemidir. Doğrudan erişimli kütük aygıtları ile donatıldığında indeks sıralı erişim metodu (Indexed Sequential Access Method-ISAM) olarak bilinir.(Güder, a.g.e., s. 227)

mümkün olduğunca arama düzenine yakın olarak kaydedilir. Hızlı bir rastgele erişim sağlamak için de indeks yapıları kullanılır. Kullanılan iki tür indeks tipi vardır.

1-Yoğun indeks (dense index) : Bu indekslerde, her arama anahtarı değeri için bir indeks kaydı bulunmaktadır. Kayıt hem arama anahtarı değeri hem de kayda ait göstergeyi içermektedir.

2-Seyrek indeks (sparse index) : Seyrek indeksler dosyadaki bazı kayıtlar için oluşturulmakta, kayda ulaşmak için aranılan arama anahtar değerine eşit veya daha küçük olan indeks kaydının bulunması gerekmektedir. Arzu edilen kayıt, indeks kaydı tarafından işaret edilen kayıttır.

Genellikle yoğun indeksler bir kayda konumlanmak için seyrek indekslerden daha hızlı çalışmaktadır. Bununla beraber seyrek indekslerin daha az alan gerektirmeleri, bilgi ekleme ve silmede verimsizliklerinin az olmasından dolayı yoğun indekslere göre avantajları bulunmaktadır. Sistem tasarımcısının erişim zamanı ve boşluk kullanımı arasında bir seçim yapması gereklidir. Her ne kadar seçime karar vermek özel uygulamaya bağlı ise de blok başına bir indeks girişine sahip seyrek bir indeks de kullanılabilir. Bu tasarım, veritabanı işlemlerinde baskın maliyetin yapılan bir isteğin disk bloklarından ana hafızaya taşınması zamanı oluşturduğundan iyi bir seçimdir. Seyrek indeksler kullanılarak aranılan kaydı içeren blokların yeri tesbit edilebilir. Kayıt bir taşma bloğu⁶⁶ üzerinde değilse indeks büyüklüğünü mümkün olduğunca küçük tutarak - dolayısıyla boşluk alanlarını azaltarak - blok erişimleri minimum yapılır⁶⁷.

İndeks kullanılıyor olsa da indeksin kendisi etkin işlemler için çok büyük olabilir. Uygulamada her bir blokta 10 kayıt içeren 100000 kayıtlı bir dosya olabilir. Blok başına bir indeks kaydı varsa indeks 10000 kayıtlı olacaktır. İndeks kayıtları veri kayıtlarından daha küçük olduğundan bir blokta 100 indeks olduğu varsayılrsa bile 100 bloklu bir indeks var demektir. Şayet indeks ana hafızada tutulabilecek kadar yeterince küçükse arama zamanı düşük olacaktır.

Eğer indeks, çok büyükse araştırma değişik disk bloklarının okunmasını gerektirecektir. İndeks b kadar blok işgal ediyor ve ikili arama⁶⁸ kullanılıyorsa en az $1 + \log_2(b)$ sonucu kadar blok okunacak demektir. 100 blok içeren bir indeks için bu yaklaşık 7 blok etmektedir. İndekslerde taşma blokları kullanılmışsa ikili aramanın kullanılması mümkün değildir. Bu durumda gerekli b bloğu okumak için sıralı erişim kullanılmalıdır. Bu tür

⁶⁶ **Taşma bloğu (overflow block) :** Doğrudan erişimli bir kütükde (direct access file) taşma kayıtlarını (overflow records) yerleştirmek için kullanılan alandır. Bkz. bucket). (Güder, a.g.e., s. 350)

⁶⁷ Korth and Silberschatz, a.g.e., p. 268-270

⁶⁸ **İkili arama (binary search) :** Herhangi bir yolla sıralanmış (örneğin alfabetik olarak) bir liste içinde belli bir elemanın aranmasında kullanılan yöntem. Bu aramada aranılan eleman listenin orta elemanı ile karşılaştırılarak listenin hangi yarısında yer almış olabileceğine karar verilir. Daha sonra bu işlem listenin seçilen yarısıyla aranılan eleman bulununcaya ta da listede olmadığına karar verilmeye kadar devam ettirilir. Bu yöntem çok hızlı çalışmasına rağmen listenin sıralı olmasının gerekliliği gib bir dezavantaja da sahiptir. (Filiz Yalçın ve Fikret Şahin, 1993, Fono Yayınları, Açıklamalı Bilgisayar Terimleri Sözlüğü, Ali Bayram (Ed.), Fono Açık Öğretim Kurumu, İstanbul, s. 20)

dosyalarda kütük boyutu büyüdükçe aranan kaydı bulmak için gerekli olan bölme ve karşılaştırma işlemi sayısı arttığı ve bir o kadar da okuma işlemi gerektirdiğinden bu yöntem giderek etkinliğini kaybetmektedir⁶⁹. Bu yöntem "dichotomizing search" olarak da bilinmektedir⁷⁰.

Bu problemi çözmek için seyrek indeks üzerinde bir indeks oluşturulur. Kayda konumlanmak için öncelikle dış indeks üzerinde ikili arama yapılarak arzu edilene eşit veya daha küçük arama anahtarı tesbit edilir ve bu işlem arama anahtarı bulununcaya kadar sürer. Kayıttaki gösterge aranılan dosyadaki kaydı gösteren bloğu işaret etmektedir. Dış indeks halen ana hafıza üzerinde ise, iki seviyeli indeksleme kullanılarak 7 yerine sadece bir indeks bloğu okunacaktır. Dosya son derece büyükse dış indeks de büyüyeceğinden ana hafızaya sığmayacaktır. Bu durumda derhal başka bir seviyede indeks -gerektiği kadar- oluşturulur. Uygulamalarda genellikle iki seviye yeterli olmaktadır. Üçten daha fazla olmasına nadiren rastlanır. Herbir indeks seviyesi genellikle fiziksel bir depolama ünitesine karşılık gelmektedir. Dolayısıyla indekslerle iz, silindir v.b. disk seviyelerine ulaşılmış olunmaktadır. Ne tür bir indeks kullanıldığına bakılmaksızın her indeks dosyaya kayıt eklendiği veya dosyadan kayıt silindiği zaman mutlaka güncelleştirilmelidir. Tek seviyeli indekslerde güncelleştirme algoritmaları aşağıda verilmiştir.

1.17.1. Silme Algoritması

Kaydı silmek için önce silinecek kayıt aranmakta, bu kayıt özel arama anahtar değerine sahip ise indeksten arama anahtar değeri silinmektedir. Yoğun indekslerde arama anahtarı değerini silmek dosyadan bir kayıt silmeye benzer. Seyrek indekslerde indeksteki arama anahtar değeri -arama anahtar sırası- indeksin başlangıcına yerleştirilmekte, sonraki arama anahtar değeri indeks girişinde varsa giriş silinir⁷¹.

1.17.2. Araya Ekleme Algoritması

Araya eklenecek kayıt değeri araştırılıp arama anahtar değeri indekste aranır. İndeks yoğun ise ve arama anahtar değeri indekste görünmüyorsa araya ekleme yapılabilir. Seyrek indekslerde ise yeni bir blok oluşturulmadıkça değişiklikler yapılamaz. Bu durumda yeni blokta görülen ilk arama anahtar değeri veritabanına aktarılır. Standart bir sıralı indeks dosyasında sadece bir indeks tutulmaktadır. Şayet farklı arama anahtarı kullanan değişik indeksler varsa, dosyanın sıralı erişimini tayin eden indeks arama anahtarı birincil indeks olmaktadır. Diğer indeksler ikincil (secondary) indeks olarak bilinir. Birincil indeksin arama anahtarı birincil anahtardır. İkincil indeksler birincil indekslerden farklı bir şekilde dizayn edilmektedir. Göstergeler doğrudan dosyayı işaret etmemekte bunun yerine dosyaya göstergeler içeren bir bucket⁷² ile işaret edilmektedir. Bu yaklaşım ikincil arama anahtar değerleri için tüm göstergelerin bir arada saklanmasını gerektirmektedir. Daha çok göstergeler kullanılarak yapılan kesin sorgulamalar için kullanışlıdır.

⁶⁹ Korth and Silberschatz, a.g.e., p. 271

⁷⁰ Güder, a.g.e., s. 41

⁷¹ Korth and Silberschatz, a.g.e., p. 271

⁷² **Bucket** : Özellikle verilerin biriktirilmesi için ayrılmış olan bellek parçasını belirtmek için kullanılan argo bir deyimdir. Daha çok başlangıç planlaması yapılırken kullanılır. (Güder, a.g.e., s. 50)

Sıralı bir aramada birincil anahtar sırası yeterlidir. Çünkü fiziksel olarak saklanan kayıtlar, birincil anahtar sırasındadır. Hem birincil hem de ikincil anahtara dayalı dosyalar fiziksel olarak saklanamamaktadır. Çünkü ikincil anahtar sırası ile fiziksel anahtar sırası farklıdır.

İkincil anahtar sıralı bir dosya, sıralı olarak taranmaya kalkılırsa her bir kaydın okunması için diskten yeni bir blok okunması gerekecektir. Göstergeler bir bucket'de saklanarak ilave göstergelere duyulan ihtiyaç azaltılmış ve ikincil anahtar sırasında yapılacak bir tarama ortadan kaldırılmış olmaktadır. Birincil anahtardan ziyade ikincil indeksler kullanılarak yapılan sorgulamalar performansı daha da artırmaktadır. Fakat veritabanı modifikasyonlarında ciddi verim düşüklüklerine tesir etmektedirler. Sıralı indeks dosyalarının, kütüklerin büyük olmaları halinde performans düşüklüğüne yol açtıkları bilinmektedir⁷³.

1.18. Sorgu İşleme (Query Processing)

Veritabanındaki veri yapısının nasıl düzenleneceğine dair verilen kararlar veritabanı tasarlanırken yapılmalıdır. Her ne kadar bu yapılar üzerinde değişiklik yapmak mümkün ise de bu işlem nisbeten verimsiz olmaktadır. Dolayısıyla bir sorgu sisteme iletiildiği zaman, mevcut veritabanı yapısı kullanılarak cevap için en uygun metodun bulunması gerekmektedir. Bir sorguyu işleyen değişik stratejiler mevcuttur. Sistem açısından önemli olan bir stratejinin seçimi için harcanan gerçek zamandır. Tipik olarak strateji seçimi çok az veya hiç disk erişimi olmaksızın hafızadaki bilgiler kullanılarak yapılabilmektedir. Sorgunun hakiki çalışması ise diske birkaç defa erişimi gerektirmektedir.

Örneğin SQL'de bir sorgu değişik yollarla belirtilebilmektedir. Yolların herbiri cevabı bulmak için farklı bir strateji gerektirmektedir. Bu strateji sistemin sorumluluğunda olup sorgu kullanıcı tarafından girildiği vakit daha etkin bir şekilde hesaplanacak eşit bir sorgu içine aktarılmaktadır. İyileştirme, daha doğrusu bir sorgu için strateji geliştirme, sorgu iyileştirme (query optimization) olarak adlandırılır. Bir derleyicinin yaptığı kod iyileştirme ile veritabanının yaptığı sorgu iyileştirme arasında yakın bir benzerlik vardır. İyi ve kötü bir strateji seçimi arasındaki çalışma zamanı farkı büyük olabildiğinden veritabanı sistemleri için bu işlem çok önemli bir görev olmaktadır.

Şebeke ve hiyerarşik modellerin çoğunda sorgu iyileştirme, uygulama programcısına bırakılmıştır. Veri yönetim dili ifadeleri ana program içerisinde saklı olduğundan tüm uygulama programı hakkında bilgi sahibi olmadıkça şebeke veya hiyerarşik bir sorguyu, ona eşit bir sorguya çevirmek kolay değildir. Kullanışlı optimizasyon tekniklerinin çoğu ilişkisel bir modele uygulanabilmektedir. Sistem, sorgu işlemeye başlamadan önce sorguyu kullanılabilir bir forma çevirmektedir. Bu çevirme işlemi aynen bir derleyici gibi çalışmaktadır. Şöyle ki; sorgunun dahili formunu üreten işlem kullanıcı sorgusunun yazımını gramer açısından kontrol eder, sorguda isimleri geçen ilişkileri -veritabanındaki ilişki adlarını- doğrular. Sorgu ifadesi bir görüntüde tanımlanmışsa, gramer kontrolünü yapan birim (parser) tüm referansları ve hesaplanacak ilişkisel cebir deyimlerini görüntü adına yerleştirmektedir. Sorguyu bundan sonra iyileştirme işlemi beklemektedir. İyileştirmenin ilk safhası ilişkisel cebir seviyesinde yapılmaktadır. İkinci adım sorgu için detaylı strateji seçimidir. Sorgunun nasıl çalıştırılacağına seçimi mutlaka tam olarak

⁷³ Korth and Silberschatz, a.g.e., p. 271-297

yapılmalıdır. Ayrıca bu aşamada özel indekslerin ve işlenecek satırların sırası da seçilmektedir. Son adım ise gerekli disk erişimlerinin sayısı üzerinde bir strateji seçimidir. İlişkisel cebir yapısal bir dil olduğu için bunlar mümkün olduğunca en hızlı bir şekilde operasyonları seçip icra etmektedirler.

Diske erişim ve okuma/yazma işlemlerinin yapılması anında bazı ara sonuçlar üretilmektedir. Geçici sonuçları azaltmak için seçimlerin önceden yapılması, birleştirme (join) operasyonlarının optimal olarak seçimi, operasyonların projelendirilmesi gibi teknikler uygulanabilmektedir. Operasyonların projelendirilmesinin, ilişkilerin hacmini azaltmak ve ihtiyaç duyulduğu zaman geçici ilişkilerin oluşturulup mümkün olan herhangi bir uygulamaya tatbik edilebilmesi gibi avantajları vardır. Bir sorgu için seçilen strateji her bir ilişkinin büyüklüğüne ve kolonlar içindeki değerlerin dağılımına bağlıdır⁷⁴.

Güvenilir bir strateji seçebilmek için veritabanı sistemleri her bir ilişki için istatistiki bilgiler saklarlar. İstatistiklerde ilişki r 'deki satırların sayısı, sabit uzunluklu kayıtlar için byte olarak ilişki r 'nin bir kaydının büyüklüğü, ilişki r 'de görülen belirli bir özellik için farklı değerlerin sayısı gibi bilgiler bulunmaktadır. Gerçek manada istatistiklerin sürdürülebilmesi için, bir ilişkide yapılan modifikasyonlardan dolayı istatistiklerin mutlaka güncelleştirilmesi gerekmektedir. Bu da gerçekten önemli bir verim düşüklüğüne yol açmaktadır. Zaten çoğu sistemler de bu yüzden her modifikasyon için istatistikleri güncellememekte, sistem yükünün hafif olduğu periyotlarda bu işlemi yapmaktadırlar. Sonuçta seçilen ve sorgu işlem stratejisince kullanılan istatistikler doğru bilgiyi yansıtmamaktadır. Güncelleştirme aralığı çok uzun değilse istatistikler nisbeten doğru bilgileri yansıtmaktadırlar. İlişkisel cebir için düşünülen maliyet tahminleri indekslerin değil değerlendirilen deyimlerin maliyetidir. İndekslerin yokluğu sorgu işlem stratejisinin seçiminde önemli tesirlere sahiptir. Bir kere indeksler, indeks anahtarları üzerinde özel bir değer içeren bir kayda hızlı erişimi temin ederler. İkinci olarak, dosya kayıtlarının sıralı olarak okunmasını sağlarlar. Dosyadaki kayıtların etkin bir şekilde sıralı olarak okunmasının fiziksel sırada birbirine yakın olmaları gerektirmesi indeksler bölümünde anlatılmıştı. Şayet bir indeks bu forma uymuyorsa kümelenmiş indeks (clustering index) adını alır. Sorgu işlemi için detaylandırılan strateji, bir erişim planı olup, sadece icra edilecek ilişkisel operasyonları değil kullanılabilecek indeksleri ve operasyonların erişebileceği satırların sırasını da ihtiva etmektedir. İndekslerin kullanımı, indeks içeren bloklara erişimi gerektirdiğinden maliyet hesaplarında bu bloklara yapılacak erişimlerin de hesaba katılması gerekir. Optimal strateji seçimlerini etkileyen çeşitli faktörler vardır. Normal bağlantı gerektiren sorgular indekslerin varlığına ve ilişkiler için kullanılan fiziki depolamanın hacmine bağlı olarak değişik yollardan da işlenebilmektedir. İlişkinin satırları fiziksel olarak bir arada ise blok yönelimli bağlantılı (block-oriented join), sıralı ilişkiler mevcutsa birleşme bağlantılı (merge-join) stratejiler kullanılmaktadır. Bağlantı işleminden önce ilişkileri sıralamak, geçici indeksler oluşturmak da yararlı olabilmektedir. Strateji seçim görevinde her sorgunun değişik alt sorgulara ayrılması, strateji seçimini basitleştirmekle kalmaz çeşitli defalar rastlanan bir sorguyu özel alt sorgu iyileştiricisinin tanınmasını da sağlar. Böylece alt sorgular bir kez icra edilerek hem sorgu iyileştirme

⁷⁴ Korth and Silberschatz, a.g.e., p. 301-319

safhasından ve hem de sorgu çalışma zamanından tasarruf edilmektedir. Ticari sistemlerin çoğu oldukça karmaşık sorgu iyileştiricileri kullanmaktadırlar.

1.19. Arıza Giderme (Crash Recovery)

Diğer mekanik veya elektriksel cihazlar gibi bilgisayar sistemleri de arızalara maruz kalmaktadırlar. Arızaların çeşitli sebepleri vardır. Bunlar disk üzerinde bulunan bilgi kayıplarına sebep olan disk arızaları, genel amaçlı kaydedicilerde (register) ve ana hafızada depolanan bilgi kayıplarına yol açan donanım arızaları ve hatalı sonuçlar üretildiği için kullanıcıya sunulan sonuçların hatalı olmasına, veritabanını tutarsız hale getiren yazılım hataları da olabilmektedirler.

Yukarıdaki durumların herbirinde veritabanı sistemini ilgilendiren bilgi kaybıdır. Veritabanı sisteminin önemli bölümlerinden birisi de arızaların tesbitinden ve veritabanının geriye yüklenerek arızanın oluşumundan önceki tutarlı yapıya oturtulmasından sorumlu olan bir kurtarma/onarma şemasıdır. Oluşan arızaların herbirinin farklı bir tarzda giderilmesi gerekmektedir. Herhangi bir sistemde oluşabilecek hata türleri ise şunlardır⁷⁵:

1-Mantıksal hatalar (logical errors) : Kaynak limitlerinin aşılması, taşma (overflow), verinin bulunamaması, hatalı girişler gibi dahili şartlar yüzünden programın normal çalışmasını sürdürememesine sebep olan arızalardır.

2-Sistem hataları (system errors) : Sistemin deadlock⁷⁶, parity error v.b. gibi arzu edilmeyen bir duruma girip programın normal çalışmasını engelleyen arızalardır.

3-Sistem çökmeleri (system crash) : Herhangi bir donanım veya yazılım hatası sebebiyle tüm sistemin çalışamaz duruma gelmesidir.

4-Disk arızaları (disc failure) : Kafa çarpmaları sonucu fiziksel blokların tahrip olması veya veri transfer operasyonunun yapılamayıp disk bloklarının içeriklerini kaybetmesi ile oluşan arızalardır.

Disk arızalarının giderilmesi için depolama ortamlarının yapıları bilinmelidir. Depolama ortamları çeşit çeşit olup hız, kapasite ve arızalara karşı esnek olup olmadıklarına göre farklılık arzederler.

1-Uçucu bellek (volatile storage) : Depolanan bilgilerin, sürekli güç beslemesi olmadıkça saklanamadığı bir bellek ortamıdır. Sistem arızalarında bu bellek üzerindeki bilgiler faaliyetlerini sürdüremezler. (Ana ve yardımcı bellek gibi.)

⁷⁵ Korth and Silberschatz, a.g.e., p. 319-325

⁷⁶ **Karşılıklı kilitleme (deadlock)** : Çoklu programlamada dinamik bellek tahsisi için yapılan isteklerde iki veya daha çok programın isteklerinin çakışması sonucu olarak sürekli bekleme durumunda kalmalarıdır. **Örneğin**; program1 A cihazını kullanıp B cihazı için istekte bulunduğunda, Program2 B cihazını kullanıp A cihazı için istekte bulunmuşsa her iki program sürekli beklemek durumunda kalacaktır. Bu durum ikiden fazla cihaz için düşünülecek olursa iş içinden çıkılamayacak derecede karmaşıklaşır. Bu problemi gidermek için bulunan veya kullanılan teknikler hali hazırda yeteri kadar etkin çözüm getirememişler ve çekici değillerdir. (Güder, a.g.e., s. 122)

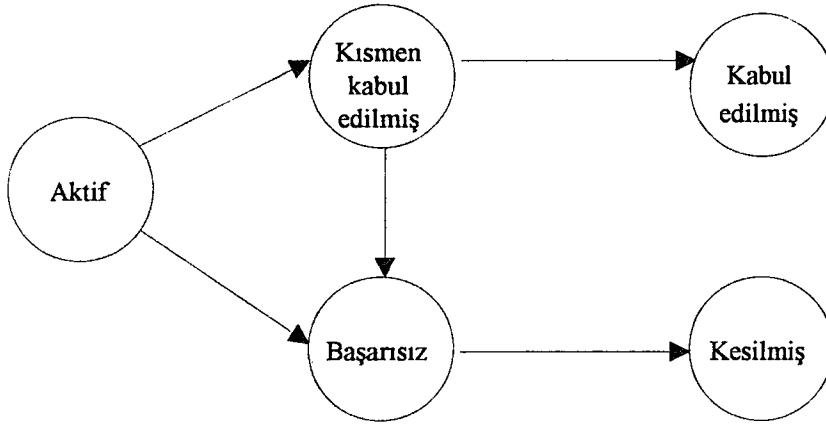
2-Uçucu olmayan bellek (non-volatile storage) : Güç kesildiğinde bile üzerindeki bilgileri tutan, kaybetmeyen, ekipmanları besleyen güç tekrar sağlandığında bu bilgileri hazır halde veren depolama ortamlarıdır. Sürekli (permanent) bellek ile eşanlımlı, uçucu bellekten farklıdır. Disk ve manyetik bantlar gibi. Diskler ana hafızadan daha güvenilir, manyetik bantlardan daha az güvenlidir. Manyetik kafa düşmesi (head crash) -disk yüzeyi ile okuma/yazma kafasının birbirine sürtünmesi- gibi arıza hallerinde bilgi kayıpları büyük miktarlara ulaşabilmektedir.

3-Durgun bellek (stable storage) : Bu cihazlarda depolanan bilgi asla kaybolmaz. Böyle bir birimin işler hale getirilmesi için bilgi uçucu olmayan, arıza durumundan bağımsız depolama ortamlarında ikinci nüshalarının oluşturulup bilginin kontrollü bir şekilde güncelleştirilmesi gerekmektedir. Bu konu üzerindeki nazari çalışmalar sürmektedir.

Bir bankacılık sisteminde A'nın hesabından B'nin hesabına 50.000 TL'lik bir transfer işlemi yapan bir programın çalıştığını düşünün. A'nın hesabında 1.000.000 TL., B'nin hesabında ise 2.000.000 TL. olsun. Programın çalışmasıyla A ve B hesaplarında sırayla 950.000 TL. ve 2.050.000 TL. olacaktır. P programının başarılı bir şekilde uyulanmasını engelleyen bir aksaklık oluşursa, daha açık bir ifadeyle A'ya ilişkin çıktı işleminin tamamlanıp, B'ye ilişkin çıktı işleminin gerçekleşmediği göz önüne alınırsa A ve B'ye ilişkin hesapların son durumu 950.000 TL. ve 2.000.000 TL. olacaktır. Yani bu aksaklık neticesinde veritabanı tutarlı bir halde olmayacaktır. Böyle bir duruma kararsız durum denmektedir. Bu da bizi -veritabanının uyumunu koruyan bir ünite programı olan- işlem (transaction) kavramına götürmektedir. Uygulama ne zaman başlarsa başlasın veritabanının kararlı bir durumda kalması arzu edilir. Bunun için de uygulama atomik -uygulamanın kullandığı tüm komutların ya tamamlanmak üzere çalıştırılması veya hiçbirinin çalıştırılmaması- bir yapıda olmalıdır.

Veritabanı bütünlüğünü koruyabilmek için çeşitli programları tanımlamak programcılarının sorumluluğundadır. İşlem ise değişik veri parçalarına erişen ve onları güncelleştiren bir programdır. Atomik yapıyı yeterince sağlayamayan işlem başarısız (aborted) işlemdir. Böyle bir uygulamanın veritabanı üzerinde hiçbir etkisinin olmaması gerekmektedir. Dolayısıyla veritabanının, problemlili uygulamanın çalışmaya başlamadan tam önceki durumuna iadesi lazımdır. Böyle bir uygulamaya da geri döndürülmüş (rolled back) uygulama denmektedir. Tamamlanmış (committed) bir uygulama ise, işlemlerini başarıyla tamamlamış demektir. Bir işlem mutlaka aşağıdaki durumlardan birinde olabilmektedir⁷⁷.

⁷⁷ Korth and Silberschatz, a.g.e., p. 325-332



Şekil 9. İşlem Durumları

- 1- Aktif, yani başlangıç durumunda
- 2- Kısmen kabul edilmiş, işlemdeki son ifadeye ulaşılması durumunda
- 3- Başarısız, donanım veya mantık hataları sebebiyle işlemin normal çalışmasının artık sürmediğinin anlaşılması durumunda
- 4- Kesilmiş, işlemin çalışması geri alınıp veritabanının işleminin başlamasından önceki durumuna getirilmesinden sonraki durumda
- 5- Kabul edilmiş, işlemin başarıyla tamamlanmış olması durumunda

Bir işlem aktif olarak başlamaktadır. Son ifadeye ulaşınca kısmen kabul edilmiş moduna girer. Bu noktada çalışmasını tamamlamış, fakat gerçek çıktı henüz diskte bulunan çıktı olmadığından kesilme ihtimali sürmektedir. Bir donanım hatası onun başarıyla tamamlanmasını engelleyebilir. İşlem başarısız durumuna, normal çalışmasının donanım veya mantıksal hatalar yüzünden işlemlere cevap verememesi halinde girmektedir. Bu tür bir işlem mutlaka geri döndürülmelidir. Geri döndürme işlemi başarılı ise kesilmiş moduna girmektedir. Bu durumda sistemin iki seçeneği bulunmaktadır. Ya işlemin dahili mantığı yüzünden değil de bazı donanım ve yazılım hatalarından kaynaklanan işlemi yeniden başlatacak veya dahili mantık hataları sebebiyle uygulama programını yeniden yazmak üzere işlemleri bitirecektir (kill).

İşlemin kabul edilmiş durumuna gelmesi için kısmen kabul edilmesi ve kesilme durumuna asla gelemeyeceğinin garanti edilmesi gerekmektedir. Atomik bir yapıyı sağlayan üç farklı şema vardır. Bunlar :

1-Güncelleştirmeleri erteleyip günlükte⁷⁸ tutmak : Bir işlemin çalışması süresince tüm yazma operasyonları işlem kısmen kabul edilinceye kadar ertelenir. Tüm güncelleştirmeler sistem günlüğüne kaydedilir. İşlem kısmen kabul edildiğinde günlükte bulunan bilgi

⁷⁸ **Günlük (Log/Journal)** : Bir bilgisayarda gelişen, oluşan veya olan tüm olay ve süreçlerin (process) kayıdır. Bu şekilde tutulan bir kayıt özellikle oluşan hataları, oluşumlarını ve sebeplerini ortaya çıkarmak, yıkılan, tahrip olan sistemi eski durumuna getirmek için gereklidir. Kayıt işlemi bir kütüğe de yapılabilmektedir.(Güder, a.g.e., s. 253)

yazma işleminin ertelendiği çalışmada kullanılan işlemle birleştirilir. İşlemin tamamlanmasından önce oluşan bir arıza durumunda günlük üzerindeki bilgi gözardı edilir. Bu işlem için daha ziyade kontrol noktalarının (checkpoint) tesbit eden teknikler kullanılmaktadır.

2-Güncelleştirmeleri yapıp g nl kte tutmak : T m g ncelle tirmeler dođrudan veritabanına ve ikinci bir g nl đe uygulanır. Arıza olu umunda birincil g nl k kullanılarak sistem  nceki tutarlı durumuna getirilir.

3-G lge sayfalama (shadow paging) : Bir i lemin hayatı boyunca iki tablo sayfası tutulmaktadır. Aktif ve g lge sayfa tabloları. Her iki sayfa da aynı yapıdadır. G lge sayfa i lemin devamı s resince asla deđi mez. Aktif sayfa tablosu, i lem bir yazma operasyonu yaptığında deđi mektedir. T m giri / ıkı  i lemlerini disk  zerindeki veritabanı sayfalarına yerle tirmek i in aktif sayfa kullanılır. I lem kısmen kabul edildiđinde g lge sayfa silinir. Aktif sayfa yeni bir sayfa olur. I lem kesilirse aktif tablo silinir. Bu t r problemlerle kar ıla mamak i in veritabanı sistemi periyodik olarak, gerekirse her g n i in ayrı ayrı, yedeklenmelidir⁷⁹.

1.20. E zamanlı Kontrol (Concurrency Control)

Modern sistemlerin en  nemli kavramlarından birisi de   phesiz  oklu programlamadır⁸⁰. I lemcinin payla ılması ile aynı anda deđi ik i lemler yapılabilmektedir. Daha az zamanda daha  ok i lemin yapıldığı bu t r bir uygulama t m bilgisayar sisteminin performansını artırmaktadır. Genel olarak bir i lem, bazı giri / ıkı  i lemlerini tamamlamak i in bekleme konumuna gelinceye kadar  alı tırılmaktadır. Tek bir i lemin yapıldığı bilgisayar sistemlerinde, i lemci tabir yerindeyse bo  bo  oturmaktadır. Bekleme s relerinin  ođu israf olmakta, hi bir faydalı i  yapılamamaktadır.  oklu programlama ile bu zaman etkin bir  ekilde kullanılmaya  alı ılmaktadır. Bir i lem beklediđinde, sistem, i lemciyi bu i lemden uzakla tırıp ba ka bir i leme tahsis etmektedir. Bu sistemin faydaları arasında i lemcinin tam kapasite ile kullanılması ve toplam i  yapma zamanını artırması sayılabilir.  oklu programlamayı tamamlayan iki yapı vardır. Etkile imli olmayan (non-interactive), toplu i  (batch) sistemleri ve etkile imli (interactive), zaman payla ımlı (time-sharing) sistemler. Etkile imli sistemlerde, i lemci  ok hızlı bir  ekilde zamanı kullanıcılar arasında payla tırdığından, herbir kullanıcı sanki kendi bilgisayarında  alı ıyormu  gibi g r nmektedir. Ger ekte ise tek bir bilgisayar bir ok kullanıcı arasında payla ılmaktadır. Veritabanında  e itli i lemler, e zamanlı olarak  alı tıđında ise veri b t nl đ  uzun s re korunamamaktadır. Sistem i in  nemli olan, e zamanlı i lemler arasında etkile imi kontrol etmek ve bu kontrol   e itli e zamanlı kontrol mekanizmalarıyla sađlamaktır. E zamanlı bir sistemde tutarlılıđı sađlamak i in serile tirme adı verilen bir mekanizma kullanılmaktadır⁸¹.

⁷⁹ Korth and Silberschatz, a.g.e., p. 333-348

⁸⁰ ** oklu programlama (multiprogramming) :** Tek bir merkezi i lem birimine sahip bir sistemde birden fazla programın belleđe getirilerek  alı tırılması, sistem kaynaklarının bu programlar arasında payla tırılması. (Filiz Yal ıner ve Fikret  ahin, a.g.e., s. 141). Bu y ntem ana bellek par alarını (partition) kullanır. (G der, a.g.e., s. 313)

⁸¹ Korth and Silberschatz, a.g.e., p. 355-397

1.21. Dağıtık Veritabanları

Dağıtık veritabanı sistemlerinde veritabanları ayrı ayrı bilgisayarlarda saklanmakta, bu yönüyle -verinin çeşitli yerleşimlerde saklanması- merkezi veritabanlarından ayrılmaktadır. Bu tür bir sistemdeki bilgisayarlar arası iletişim, yüksek hızlı yollar (bus) veya telefon hatları gibi çeşitli iletişim ortamları ile sağlanmaktadır. İşlemciler kullandıkları amaçlara göre site, node gibi farklı isimlerle anılmaktadırlar. Siteler hem yerel, hem de geniş çaplı muameleleri yapabilmektedirler. Fiziksel olarak siteler birbirlerine değişik yollarla bağlanmaktadır. Durumlarına uygun çeşitli topolojiler⁸² grafiksel olarak Şekil 10.'da gösterilmiştir⁸³. Bu konfigürasyonlar arasındaki farklılıklar kuruluş, iletişim maliyeti, güvenilirlik ve kullanılabilirlikten kaynaklanmaktadır. Siteler geniş veya küçük bir coğrafi alana yayılmış olabilirler. Geniş coğrafi alana yayılmış olan ağ tipi geniş mesafeli alan ağları (Wide Area Network-WAN), küçük alanlarda faaliyet gösteren tipleri ise yerel alan (Local Area Network-LAN) ağları olarak bilinmektedir. Uzun mesafeli iletişim ağları yerel alan ağları ile karşılaştırıldığı zaman nisbeten yavaş ve daha az güvenilirdir. Bu tür ağ bağlantıları telefon hatları, mikrodalga ve uydu kanalları ile sağlanmaktadır. Yerel ağlardaki siteleri birbirine bağlayan hatlar ise daha hızlı ve hata oranları daha düşüktür. Yaygın olarak kullanılanları UTP, baseband⁸⁴, broadband⁸⁵ eş eksenli (coax) kablo ve fiber optik kablolardır. Dağıtık veritabanı sistemlerinin kurulm amaçları arasında verinin paylaşımı, güvenilirlik, kullanılabilirlik, sorgu işlemlerini hızlandırma v.b. sayılabilir. Avantajlarına karşın bir takım dezavantajları⁸⁶ da bulunmaktadır⁸⁷.

⁸² **Ağ yapısı (topoloji) :** Veri iletişim ağının fiziksel dağılımını ve düğümler (node) arasındaki ilişkileri gösteren yapı. Ağ mimarisi olarak da bilinir. (Güder, a.g.e., s. 514)

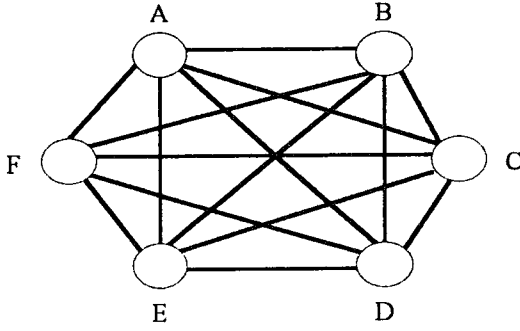
⁸³ Korth and Silberschatz, a.g.e., p. 403

⁸⁴ **Baseband :** Sinyal iletiminin modülasyona -digital sinyali analoga veya tersi- tabi tutulmadan iletilmesi. (Fitzgerald, J., 1993, Business Data Communications, Basic Concepts, Security and Design, John Wiley & Sons, Inc., Singapore, Fourth Edition, ISBN : 0-471-59983-2, p. 176)

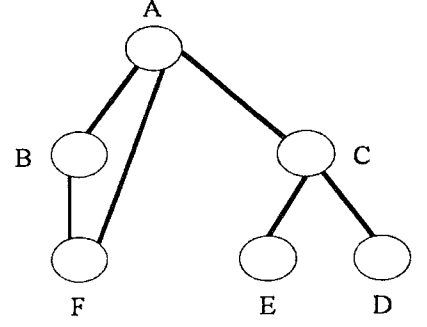
⁸⁵ **Broadband :** Olağan ses gönderme kanallarından daha geniş bir banda ve böylece daha hızlı bilgi gönderme sığasına sahip olan iletişim kanalı. (Fitzgerald, J., a.g.e., p. 521-522)

⁸⁶ **Bug/Debug :** Tam karşılıkları böcek ve böcekten arındırmadır. Bir programın beklenmedik noktalarında hatalar varsa ve program zaman zaman yanlış çalışıyorsa o programda BUG bulunduğu ilişkin terimler kullanılır. Bu hataların bulunarak düzeltilmesi süresine DEBUG adı verilir. DOS işletim sisteminde bu işi yapan aynı isimde bir dosya mevcuttur. Bu deyimın bilgisayar terminolojisine girmesinin hikayesi ise bir hayli ilginç. MARK II elektromekanik bilgisayarı için 1950'lerde program geliştiren Grace HOPPER isimli kadın programcı hatalı çalışan programındaki problemi bir türlü bulamamaktaydı. Uzun araştırmalar sonunda bilgisayarın röleleri arasına sıkışmış bir böcek ölüstünün bütün problemlerin kaynağı olduğu anlaşılmıştı. (Can Uğur Ayfer, Ağustos 1995, Kim Korkar Hain Bilgisayardan, 3. Basım, İstanbul, Pusula Yayıncılık ve İletişim Ltd., s. 27)

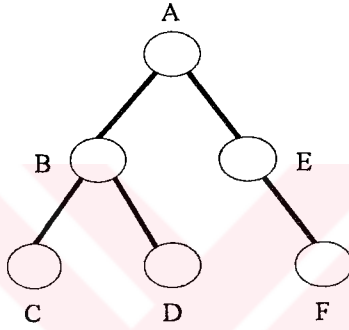
⁸⁷ Korth and Silberschatz, a.g.e., p. 404-406



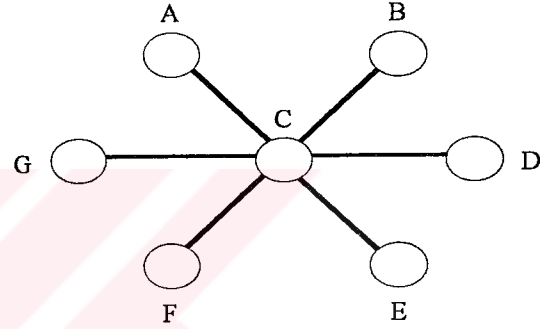
Tam Bağlantılı Şebeke
(Fully Connected Network)



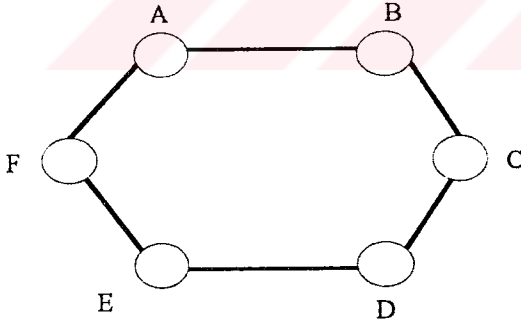
Kısmi Bağlantılı Şebeke
(Partially Connected Network)



Ağaç Yapılı Şebeke
(Tree Structured Network)



Yıldız Şebeke
(Star Network)



Halka Şebeke
(Ring Network)

Şekil 10. Ağ Topolojileri

1.21.1. Dağıtık Sistemin Avantajları

Dağıtık veritabanı sistemlerinin en önemli avantajı veriye erişim ve paylaşımında oldukça güvenilir olmalarıdır. Merkezi sistemlerde veritabanı yöneticisi, dağıtık sistemlerde ise genel bir veritabanı yöneticisi ve onun sorumluluklarını paylaşan yerel yöneticiler bulunmaktadır. Veritabanının dizaynına bağlı olarak yöneticiler farklı derecelerde bağımsızlığa (local autonomy) sahiptirler.

Herhangi bir sitenin devre dışı kalması -özellikle verinin ikinci kopyaları diğer sitelerde mevcutsa- sistemin tamamen kapanmasını gerektirmez. Hata giderme, merkezi sistemlerden daha komplekstir. Çeşitli sitelerdeki verilerin sorgulanması gerekiyorsa sorgu, alt sorgulara bölünerek bir kaç site tarafından da çalıştırılabilmektedir. Bu tür paralel hesaplamalar kullanıcı sorgusunun hızlı olarak işlenmesine mücadele etmekte ve veri birkaç siteye dağılmışsa sorgu sistem tarafından yükü en az olan siteye iletilmektedir.

1.21.2. Dağıtık Sistemin Dezavantajları

Siteler arasındaki koordinasyon sağlayıcı yapıların karmaşık olması ve yazılım geliştirme maliyetini artırması bu sistemlerin en büyük dezavantajıdır. Aynı anda işlemlerin yapıldığı dağıtık bir sistemde algoritmaların uyumluluğunu sağlamak ise ayrı bir güçlük olarak ortaya çıkmaktadır.

Veritabanı sisteminin dizaynını yapan kişi tarafından, dağıtılmış verinin avantaj ve dezavantajlarında bir muvazene sağlanmalıdır. Tamamen dağıtılmış sistem dizaynları ile geniş derecede merkezi olarak dizayn edilmiş sistemler arasında yer alan bir yelpazede, dağıtık veritabanı dizaynları için çeşitli yaklaşımlar mevcuttur. Bu yaklaşımlar şunlardır :

- 1-Tekrarlama : Sistemin devamlılığını ilişkilerin ikinci nüshaları sağlamaktadır. Veri tekrarı olarak ortaya çıkan bu durumda her bir kopya farklı bir sitede saklanmaktadır.
- 2-Parçalama : İlişkiler parçalanmakta ve her bir parça farklı bir sitede depolanmaktadır.
- 3-Tekrarlama ve parçalama : İlk iki tasarımın bir kombinasyonu olup, ilişki parçalara ayrılmakta, sistemin devamlılığı böyle her bir parçanın kopyaları ile sağlanmaktadır.

Dağıtık veritabanındaki her bir veri elemanı tek bir isme sahip olmalıdır. Merkezi sistemlerde bu kolaylıkla sağlanabilmektedir. Dağıtık veritabanlarında ise değişik sitelerin farklı veri elemanları için aynı ismi kullanmamaları mutlaka sağlanmalıdır. Çözüm olarak bütün isimler merkezi bir name-server'a kaydedilebilir. Fakat name-server'de darboğazların oluşması halinde herhangi bir sitenin çalışamaz duruma gelmesi ve isimlendirme mekanizmasının merkezden kontrol edilmesinden dolayı yerel özerkliğin sağlanamaması gibi olumsuzluklara da sahiptirler. Yerel özerkliği artırmak için üretilen veri elemanı adına, site tanıtıcısının ön ek olarak kullanılması yoluna gidilmektedir. Böylece hiçbir site benzer bir isim üretmemektedir. İsimlendirme probleminin bu şekilde çözümü, yerel özerkliği sağlamaktadır fakat site tanıtıcılarının isimlerine atıf yapıldığı için ağ şeffaflığı oluşmamaktadır. Sorgulama işlemleri için verilecek cevap süresini minimum seviyede tutacak çeşitli strateji seçim teknikleri vardır. Merkezi sistemlerde ilk kriter erişilecek disklerin sayısı idi. Dağıtık sistemlerde ise geniş ölçüde ağın yapısına ve disklerin hızına bağlı olmaktadır. Siteler arasında taşınan verinin miktarı, veri blok iletiminin maliyeti ve her bir sitedeki işlemlerin hızı gibi faktörler de gözönüne alınmalıdır⁸⁸.

⁸⁸ Korth and Silberschatz, a.g.e., p. 406-422

Dağıtık bir sistemde atomik yapıyı koruma görevi işlem yöneticisinin (transaction manager) görevidir. Diğer görevleri arasında sitede depolanan verilere erişen işlem veya alt işlemlerin çalışmasını yönetmek, kurtarma işlemleri için günlük tutmak ve paralel çalışan işlemlerin uygun kontrol şemaları ile koordinelerini sağlamak sayılabilir.

İşlem koordinatörü (transaction coordinator) ise işlemlerin çalıştırılmasını başlatmakta, işlemleri alt işlemlere bölerek uygun sitelere dağıtmakta, işlemin tüm sitelerde başladığını veya kesilmişse sonuçlarını ve işlemin sona ermesini koordine etmektedir.

Dağıtık bir veritabanı sistemi, merkezi bir sistemde oluşan benzer tipteki arızalardan etkilenmektedir. Dağıtılmış bir çevreyle ilişkide bulunduğundan dolayı da ilave aksaklıklar çıkabilmektedir. Site ve hat arızaları, mesaj kayıpları, ağ parçalanmaları v.b. gibi. Bu tür hataların oluşabileceği düşüncesiyle sistemi daha güçlü kılmak için hesaplamaların devamını sağlayacak şekilde sistemin yeniden konfigürasyonu, bir hat veya işlemci onarıldığında ise kurtarma çalışmalarının başlatılması gerekmektedir. Atomik yapıyı temin etmek için tüm sitelerde çalışan bir T işleminin sonucu ile, oluşan sonuç mutlaka uyusmalıdır. T'nin işlem koordinatörü bir kabul protokolü çalıştırarak bunu sağlamaya çalışmaktadır. T işlemi ya bütün sitelerce onaylanır veya reddedilir. Kabul protokolleri arasında en yaygın olanı iki safhalı kabul protokolüdür.

1.21.3. İki Safhalı Kabul Protokolü

S_1 sitesinde bir T işlemi başlatılmış ve bu sitedeki işlem koordinatörü de C_1 olsun. T çalışmasını tamamladığında, tüm siteler T işlemini çalıştırdıklarına dair C_1 'ye bilgi verirler. C_1 o zaman iki safhalı kabul protokolünü başlatır.

1. Safha : C_1 <T hazırlık> kaydını günlüğe ekler ve onu sabit depolama alanına kaydeder. T'nin çalıştırıldığı her siteye de "kabule hazırlık" mesajını gönderir. Sitenin işlem koordinatörü tarafından alınan cevap Hayır (no) ise <T yok> kaydı günlüğe eklenir ve C_1 'ye de "T kesildi" mesajı gönderilir. Cevap Evet (yes) ise <T hazır> kaydını günlüğe ve sabit depolama alanındaki T ile ilgili tüm günlük kayıtlarına ekler. C_1 'ye ise "T hazır" mesajı gönderilir.

2. Safha : C_1 , <T hazırlık> mesajını tüm sitelerden alınca kabul, önceden belirlenmiş zaman aralığı şılmışsa reddedildiğine karar verebilir. C_1 katılan tüm sitelerden "T hazır" mesajını alınca T işlemi kabul edilebilir aksi halde T işlemi mutlaka kesilmelidir. Verilen karara bağlı olarak ya <T kabul> veya <T kesildi> kaydı günlüğe ve diske eklenir. Bu noktada işlemin akıbetine karar verilir. İşlem koordinatörü tüm sitelerde <T kabul> veya <T kesildi> mesajını gönderir. Bir site bu mesajı alınca onu günlüğe kaydeder ve <T alındı> mesajını koordinatöre gönderir. Koordinatör <T alındı> mesajını tüm sitelerden alınca <T tamamlandı> kaydını günlüğe ekler⁸⁹.

⁸⁹ Korth and Silberschatz, a.g.e., p. 422-426

1.22. Güvenlik ve Bütünlüğü Sağlama

Bilgi güvenliği, bilginin yetkili olmayan kişilerce kazara veya kötü niyetle değiştirilmesi, silinmesinin önlenmesi olarak tarif edilebilir. Bilgiyi korumak için gerekli fiziksel, örgütsel ve teknik önlemler alınmalıdır. Hangi kullanıcıların hangi dosyaları, kayıtları, alanları görebileceği, hangilerine düzeltme hangilerine silme işlemi uygulayacağı belirlenmelidir. Dış şemalar ve uygulama programları buna göre dizayn edilmelidir. Kullanıcı yetkili olmadığı bilgilere ulaşmaya çalışırsa uyarılmalı, ek şifre sorulmalı veya sistem kendini kilitleyebilmelidir. Yaygın kullanılan DBMS'lerde değişik şekilde ilave yöntemler de bulunmaktadır. Bazı kişiler bir şekilde sistemin bu kontrollerini atlatabilmektedir. Bu gibi durumlar için de bilgilerin gizlilik fonksiyonları ile şifrelenmiş (encryption) olarak saklanması bir çözüm getirebilecektir. Değişik doğruluk sınırlamalarına ek olarak dosyaların belirli periyodlarla yedeklenmesi (backup), uygulanan her işlemin kaydedilmesi (journaling), yedeklenmiş ve journal dosyalarından iyileştirme programları (recovery utility) yardımı ile en son bilgi durumunun oluşturulması v.b. gibi bir takım yöntemlerden faydalanılır.

Veritabanlarının önemli bir özelliği de verilerin paylaşımıdır. Aynı dosyayı birden çok kullanıcı aynı anda kullanabilmektedir. Özellikle güncelleme yaparken bilgilerin bozulması söz konusu olabilir. Birden çok kullanıcı aynı dosyayı kullanmaya çalışırken sistemin kilitlenmesi söz konusu olduğundan bu ve benzeri problemler önceden titizce düşünülüp, çözümlenmelidir. Veritabanında depolanan veri, yetkisiz ve kötü niyetli erişimlerden, değişikliklerden, kaza eseri ortaya çıkacak uyumsuzluklardan korunmuş olmalıdır. Uyumsuzluğun ve yanlış kullanımların veri üzerinde olumsuz sonuçlara yol açacağı unutulmamalı ve bunlara karşı alınacak tedbir mekanizmaları bilinmelidir.

Veritabanı güvenliği zararlı erişimlere karşı alınacak tedbirleri anlatırken, bütünlük uyumluluğun kazaen kaybindan korunmak anlamında kullanılmaktadır. Pratikte güvenlik ve bütünlüğü ikiye ayıran çizgi her zaman net değildir. Burada güvenlik terimi hem güvenliğin hem de bütünlüğün sağlanması anlamındadır. Veritabanını korumak için fiziksel, insan, işletim ve veritabanı seviyelerinde güvenlik önlemleri mutlaka alınmalıdır⁹⁰.

Çoğu uygulamalar için veritabanının güvenlik ve bütünlüğünü muhafaza etmek çok çok önemli olmaktadır. Bilgilerin kaybı, bu bilgilere sahip teşebbüslerin çalışmalarını sekteye uğratacaktır. Veritabanı sistemi seviyesinde, adres bazında güvenlikler de sağlanabilmektedir. İşletim sistemi güvenliğinden kastedilen çeşitli seviyeler sisteme erişim için gerekli şifreden, uyumluluk için sistemde çalışan işlemleri ayırmaya kadar uzanmaktadır. Dosya sistemi de bazı güvenlik dereceleri sağlayabilmektedir. Görüntülerin içeriği bir anlamda veritabanının kullanıcılar için "kişileştirilmesi" dir. Kullanıcının görmesine ihtiyaç duymayacağı veriler gizlenerek bir nevi güvenlik sağlanmış olur. İlişkisel veritabanı sistemleri biri ilişki, diğeri görüntü seviyesinde olmak üzere iki seviyeli bir güvenlik sağlamaktadır. İlişki seviyesinde, kullanıcının bir ilişkiye doğrudan erişimine izin verilmekte veya bu erişimleri sınırlanmaktadır. Görüntü seviyesinde ise kullanıcıya görüntüdeki veriye erişim izni verilmektedir. Kısaca ilişki ve görüntü seviyeli güvenlik

⁹⁰ Korth and Silberschatz, a.g.e., p. 449

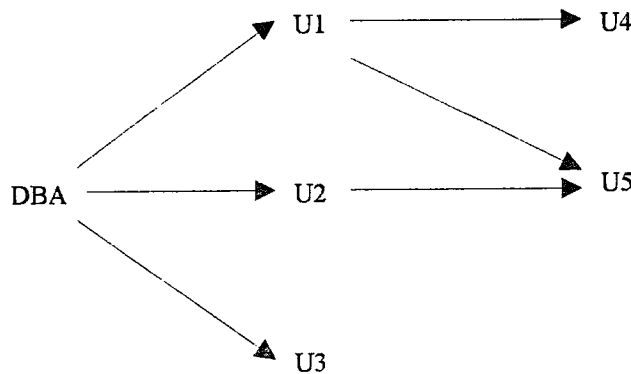
kombinasyonları, kullanıcının ihtiyaç duyduğu veriye erişimi sınırlandırmak için kullanılmaktadır. Bir kullanıcı ayrıca veritabanının bölümleri üzerinde çeşitli yetki formlarına sahip olabilmektedir. Yetki (authorization), veritabanı sisteminin art niyetli veya yetkisiz erişimlere karşı korunması anlamına gelmektedir. Bazı yetkileri garanti edilmiş bir kullanıcı bu yetkisini diğerlerinin de kullanabilmesi için onlara devredebilir. Bütünlüğü sağlamak için yetkilerin kullanıcılar arasında nasıl dağıtılacağı ve ileride geri alınabilmeleri hususunda çok dikkatli olunmalıdır. Kullanıcıların sahip olabilecekleri yetki formları şunlardır.

- 1-Okuma (read) yetkisi : Verinin okunması iznidir. Veriler üzerinde değişiklik yapılamaz.
- 2-Kayıt ekleme (insert) yetkisi : Veritabanına yeni bilgilerin eklenmesi iznidir. Mevcut verilerde değişiklik yapılamaz.
- 3-Güncelleştirme (update) yetkisi : Veri üzerinde değişiklik yapılabilmekte, fakat veri silme işlemi yapılamamaktadır.
- 4-Silme (delete) yetkisi : Verinin silinebilmesi iznidir⁹¹.

Yukarıdaki yetkilere ilave olarak veritabanı şemaları üzerinde modifikasyonların yapılabilmesi için kullanılan yetkiler de mevcuttur⁹².

- 1-İndeks yetkisi : İndeksleri oluşturma ve silebilme yetkisidir.
- 2-Kaynak (resource) yetkisi : Yeni ilişkileri oluşturabilme yetkisidir.
- 3-Değişiklik (alteration) yetkisi : İlişkileri silebilme veya ekleyebilme yetkisidir.
- 4-Düşürme (drop) yetkisi : İlişkilerin silinmesi yetkisidir.

Veritabanı yöneticisinin sahip olduğu yetki olan DBA yetkisi, sistemdeki en yüksek yetkidir. Bu yetkiyi kullanarak yeni kullanıcıları yetkilendirebilir, veritabanını yeniden yapılandırabilir v.b. Veritabanı yöneticisinin bir X tablosu üzerinde kullanıcı U1,U2,U3 'e değişiklik yetkisi verdiğini farzedelim. Bu kullanıcılar yetkilerini diğerlerine devredebilirler. Verilen yetkiler, yetki ağaçları ile gösterilmektedir.



Şekil 11. Yetki Ağacı

⁹¹ Korth and Silberschatz, a.g.e., p. 451

⁹² Korth and Silberschatz, a.g.e., p. 452

Veritabanı yöneticisi U1'in yetkisini geri almaya karar verirse U4, U1 tarafından yetkilendirildiği için bu yetki geri alınabilir. Fakat U2'nin değişiklik yetkisini geri alırsa U5, U1 tarafından da yetkilendirildiği için U5'in değişiklik yetkisi halen sürmektedir⁹³.

Yetki formlarının veritabanını zararlı veya yetkisiz erişimlere karşı koruduğundan söz etmiştik. Buna karşın bütünlük sınırlamaları farklı bir amaca, yetkisiz kişilerce veritabanına yapılacak değişikliklerin veri uyumluluğunu bozmamasına, hizmet etmektedirler. Şebeke ve varlık-ilişki modellerinde bütünlük yapısı hem anahtar bildirimlerle hem de bire-bir, bire-çok, çoğa-çok formlarıyla sağlanmaktadır. Şebeke modelinde ayrıca saklama⁹⁴ yapıları da kullanılmaktadır. SQL'in teklif ettiği genel amaçlı bütünlük sınırlamaları ise "assert" ifadesi ile temin edilmektedir. Örneğin depozit isimli bir tabloda bir hesap bakiyesinin negatif olamayacağı tanımlanacaksa şu SQL ifadesi kullanılmalıdır⁹⁵.

Assert bakiye_siniri on depozit bakiye >=0

Bir "assert" ifadesi çalıştığı zaman sistem doğruluk testi yapar. İddia doğruysa ve ileride veritabanına yapılacak modifikasyonlar bir tehlike arzetmiyorsa ifadenin çalışmasına izin verilmektedir. Bu teknik çok kompleks yapıli iddialarda zorluklar oluşturmaktadır. Bu tekniğin sakıncalarını ortadan kaldıran tetikleme⁹⁶ isimli alternatif bir şema kullanılmaktadır. Yetkilendirme için veritabanlarında yapılan çeşitli tedarikler yüksek yoğunluklu (sıkıştırılmış) olan veriyi korumak için yeterli olmayabilir. Bu gibi durumlarda veri şifrelenebilmektedir (encrypt). Şifrelenen verinin, şifrenin nasıl çözüleceği (decrypt) veriyi okuyan tarafından bilinmedikçe okunması mümkün olmamaktadır. İyi bir şifreleme tekniği şu özellikleri içermelidir.

- 1- Kullanıcıları şifreleme ve şifre çözmede nisbeten basit olarak yetkilendirmelidir.
- 2- Şifreleme şemaları, algoritmanın değil şifreleme anahtarının (encyrption key) parametrelerine bağlı olmalıdır.
- 3- Davetsiz misafirlerin şifreleme anahtarını bulmalarını son derece güçleştirmelidir.

Örneğin -kötü bir şifreleme tekniği olmakla beraber- veri karakterlerinin alfabele kendinden sonra gelen karakterle değiştirilmesi yapılabilir. Böylece "veritabanı" terimi "yfskubcbok" olacak, yetkisiz kullanıcılar sadece bunu görecektir. Bu amaçla kullanılan yaklaşım Veri Şifreleme Standardı (Data Encryption Standard -DES) olarak anılmaktadır. Şifreleme şemalarının çalışabilmesi için yetkili kullanıcılara güvenlik mekanizmaları yoluyla şifreleme anahtarları verilmelidir. Şifreleme anahtarının iletimi ise güvenlik sağlamaktan çok güvenli olmayan bir mekanizma olduğundan büyük bir

⁹³ Korth and Silberschatz, a.g.e., p. 452-455

⁹⁴ **Saklama (retention)** : Verilerin ne kadar süre ile saklanacağını ifade eden bir veri güvenliği yöntemi.(Anthony Chander with J. Graham and R. Williamson, 1985, Dictionary of Computers, Penguin Group, England, London, (4rd Edition), ISBN : 0-14-051127-X, p. 392)

⁹⁵ Korth and Silberschatz, a.g.e., p. 455-457

⁹⁶ **Tetikleme (triggering)** : Bir program veya makinayı otomatik olarak eyleme geçirmektir. Örneğin programı belleğe yükledikten sonra bellekteki programın bir giriş noktasına (entry point) otomatik olarak yapılan bir dallanma.(Güder, a.g.e., s. 523)

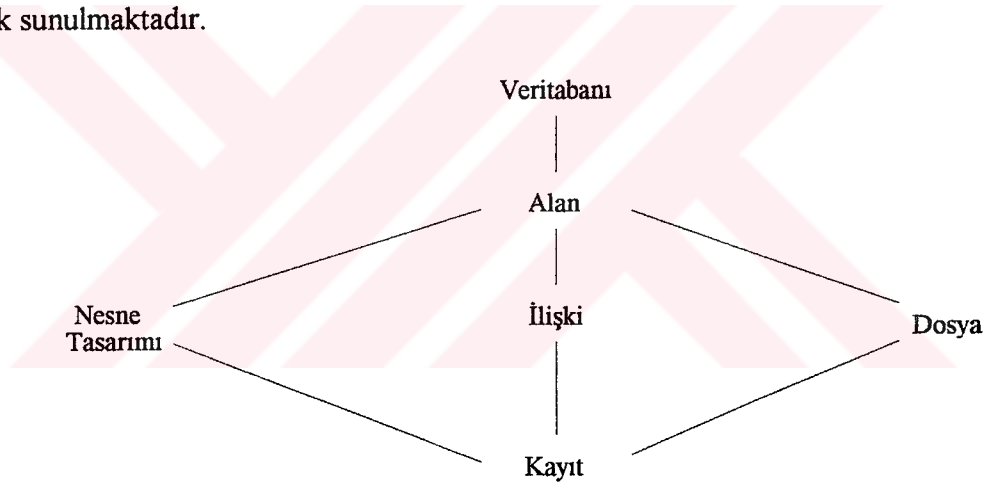
handikaptır. DES bu tür problemlerden korunmak için alternatif bir şema geliştirmiştir. Bu şema (public key encryption) olarak bilinmektedir. Biri genel (public) diğeri özel (private) olan iki anahtar üzerine tesis edilmiştir. Genel olan şifreler herkese açıktır. Özel olanları ise sadece o kullanıcı bilmektedir.

Her bir kullanıcı (U_i) genel (E_i) ve özel (D_i) anahtarlarına sahiptir. Kullanıcı (U_1) şifreli veriyi saklamak isterse genel anahtar olan (E_i)'yi kullanarak bu işlemi yapar. Şifreyi çözmek için özel anahtar olan (D_i) gerekmektedir. Kullanıcı (U_1) verilerini (U_2) ile paylaşmak isterse veriyi (E_2)'yi kullanarak şifreler. Genel anahtarı ise (U_2) dir. Çünkü sadece (U_2) kullanıcısı veriyi nasıl deşifre edeceğini bilmektedir.

Güvenlik için bir başka yaklaşım veri kirliliğidir. Bir sorguya cevap olarak sağlanan verinin gelişigüzel taklit edilmesi bu yaklaşımın felsefesidir. Benzer bir teknik sorgulamanın kendisinin gelişigüzel modifikasyonlar içermesidir. Her iki teknikde de amaç, doğruluk ve güvenliğin sağlanmasıdır⁹⁷.

1.23. Bilgi Tabanları

Yapay zeka⁹⁸ ve uzman sistemlerde⁹⁹ ise bilgi, mantıksal bir biçimde ifade edilen olaylar olarak sunulmaktadır.



Şekil 12. Nesne Tasarımı

Uzman sistemlerin¹⁰⁰ güçlü kuralları vardır. Verileri ve bilgi tabanındaki¹⁰¹ kuralları veritabanında saklamak gibi iki basit görevi kullanarak işlemler yapmaktadır. Kurallar

⁹⁷ Korth and Silberschatz, a.g.e., p. 458-461

⁹⁸ **Yapay zeka (artificial intelligent)** : Bilgisayarların, işlemlerini insanların öğrenme ve karar verme yeteneklerine benzer şekilde yapacağı bir yolla kullanılmalarını açıklamak için kullanılan bir terimdir. (Güder, a.g.e., s. 21; Chandor, a.g.e., p. 40)

⁹⁹ **Uzman sistem (expert system)** : Belli bir uzmanlık alanında kullanılmak üzere tasarlanan, insan tecrübe ve bilgisine dayalı veri ve yapay zeka programlarından oluşan sistem. Örneğin tıbbi teşhis sistemlerinde bilgisayara hastalık belirtileri (symptoms) girilerek teşhis ve tedavi şekilleri belirlenebilmektedir. (Chandor, a.g.e., p. 178)

¹⁰⁰ Bkz. J. A. Voelker, G. B. Ratica, 1986, The Genesis of a Knowledge-Based Expert System, IBM Systems Journal, Vol. 25, No 2, U.S.A., ISSN : 18-8670, p. 181-189

tipik olarak bir if-then formunda tarif edilen mantıksal ifadelerdir. Bilgi tabanlı sorgu işlemcilerinden daha güçlüdürler. Her ne kadar sorgu işlemcilerin ve uzman sistemlerin her ikisinin de asıl amacı bir sorguya cevap vermekse de, genellikle bir veritabanı sorgu işlemcisinden alınan sorguları uzman bir sisteme aktarmak asıl amaçtır. Uygulamalar genellikle problemlerin teşhisinde kullanılmaktadır.

Bilgi temelli/tabanlı bir sorgunun işlenmesi bir veritabanı sorgusunu işlemekten daha komplekstir. Veritabanındaki veriler veri değerlerinin bir topluluğudur. Bilgi temelindeki veriler ise değişkenlere ait kurallardan meydana gelmektedir. Veriler bu değişkenlere sadece sorgu işlemi esnasında atanmaktadırlar. Bu yüzden bilgi tabanlı sorgu işlemcisi, bir değişken-değer bağlayıcı listesinin koruma ve bakımını sürdürmek zorundadır. Geniş bir bilgi tabanında sorgu işlemcisi indekslere ihtiyaç duymaktadır. Bilgi tabanlarına veri ekleme, silme, güncelleştirme -bir takım güçlüklerle rağmen- yapılabilmektedir. Değişiklik işlemleri düğümdeki elemanların değerleri için ayrı ayrı özelliklerinin ve bütünlük sınırlamalarının test edilmesini gerektiren kontrol fonksiyonları aracılığı ile yapılır. Bir kural bilgi tabanına eklendiği zaman mevcut bir bilginin aksini iddia etmesi mümkündür. Bu tür değişikliklere izin verilmez. Yine bilgi temeline mevcut kurallardan türemiş kuralların eklenmesi de mümkündür. Bir bilgi temeli kurallar seti ve veri topluluklarından oluşmaktadır. Uzman sistemlerin ilk uygulamaları veri yönetimi için veritabanlarını kullanmadılar. Bunun yerine özel uygulamalara has teknikler kullandılar. Bu tekniklerin kabul edilebilir olması, bilgi temellerinin ana hafızada çalışmak için yeterince küçük olmaları ve başlangıçta tecrübeye dayalı sistemlerde performansın pek önemli olmamasından dolayıdır. Uzman sistemler, veritabanı ile sıradan bir kullanıcı gibi etkileşimde bulunup, sorgularını SQL gibi dillerle veritabanına iletirler ve bir cevap beklerler. Bu işlem kolay gibi görünse de, veritabanı sorgusunda bilgi temelindeki kuralları kullanmak pek optimal bir tasarım değildir. Bu yüzden değişik alternatifler düşünülmüştür. Uzman sistemler, mantıksal veritabanları ve yapay zeka alanındaki mevcut çalışmalar bilgi temelindeki problemleri çözmeye çalışmaktadırlar¹⁰².

1.24. Çok Ortamlı Veritabanları

İlk veritabanları sadece dosyaları saklayabilecek şekilde geliştirilmiştir. Gelişen yazılım teknolojisi ile kapasiteleri artan donanımlar sayesinde daha farklı biçimdeki bilgilerin saklanabileceği veritabanları oluşturulmaktadır. Bu modern veritabanlarında basit dosyaların yanında uzun metinler, grafik, görüntü ve ses de saklanabilmektedir. Bu tür veritabanları eğitimde, müzecilik, emlak sistemlerinde, güvenlik sistemleri, kütüphanecilikte v.b. pek çok yerde kullanılmaktadır¹⁰³.

1.25. Veritabanı Programları

Burada ya ticari olarak kullanılabilen veya önemli etkiye sahip seçilmiş birkaç veritabanı programı incelenecektir. Tüm detayların sunulması yerine bu sistemlerin birkaç özelliğine işaret edilecektir. Sunulan bilgiler ürünlerin tüm özellikleri şeklinde yorumlanmamalıdır¹⁰⁴. İlişkisel, hiyerarşik ve şebeke modellerinde en yaygın kabul

¹⁰¹ **Bilgi tabanı (knowledge base)** : Belirli bir konu ile ilgili problemlerin çözümünde kullanılacak gerçekleri, yordamları ve tüm bilgileri ihtiva eden veritabanı sistemi. (Chandor, a.g.e., p. 262)

¹⁰² Korth and Silberschatz, a.g.e., p. 473-477

¹⁰³ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 259

¹⁰⁴ Korth and Silberschatz, a.g.e., p. 481

görmüş veritabanı sistemlerinin ardından Client/Server ve Windows veritabanlarına bir göz atılacaktır.

1.25.1. İlişkisel Veritabanları

1.25.1.1. System R

Bu çalışma, IBM San Jose Araştırma Laboratuvarında 1974'de başlamıştır. Amacı o zamanlar yeni ortaya konmuş olan ilişkisel veri modelinin kullanılabilirliğini göstermektir. Bu çabanın başarısı hem bu sistemden türeyen ürünlerden hem de araştırma sonucu ortaya çıkan çok sayıda temel kavramdan dolayı açıkça görülmüştür. Bu proje 1979'da tamamlanmış ve IBM kısa bir süre sonra SQL duyumuştur. System R araştırmasının tamamı da R* dağıtık veritabanı araştırma projesinin temeli olmuştur. System R çabasının temel katkıları arasında SQL sorgu dilini kullanım, sorgu derleme ve optimizasyon, geleneksel bir programlama dili ile ilişkisel bir dilin entegrasyonunu sağlama, serileştirme ve iki safhalı kilitleme sayılabilir¹⁰⁵. System R'nin iç mimarisi iki temel parçadan oluşmaktadır.

1-İlişkisel Depolama Sistemi (İDS) : Disk depolama ve ana hafıza yönetimi, arıza giderme ve uyumluluk kontrolünü sağlamaktan sorumludur.

2-İlişkisel Veri Sistemi (İVS) : Görüntüler (view), onaylama ve bütünlükten sorumludur.

İDS, mantıksal modelde ilişkilere tek tek kayıt seviyesinde ulaşmayı mümkün kılar. Yüksek düzeyde erişim İVS arayüzü ile elde edilir. Bu arayüz SQL ifadelerini kabul etmektedir. İVS' nin üst kısmında, ortak System R veritabanına istenen sayıda arayüz inşa edilebilir. Bunların en çok kullanılanı User-Friendly Interface (UFI) arabirimidir. Bu arabirim SQL ifadelerini kullanıcıdan kabul eder, bunları İVS'ye aktarır ve sonuçları gösterir. Ayrıca yapılacak sorgular incelenip bir strateji seçimi yapmakta (precompilation), bunları veritabanında depolamaktadır. System R, sadece PL/1 ve Cobol içerisinden kullanılmaktadır. PL/1'in kayıtlar üzerinde, SQL' in de kayıt seti üzerinde işlem yapması sebebiyle bir takım güçlükler vardır.

Örneğin x, PL/1 programına ait bir değişken olsun ve "X dolardan daha fazla hesabı olan müşterilerin ad ve soyadlarını bul" sorgusunu, sonuç kaydını tutmak için de a ve b PL/1 değişkenlerini kullandığınızı farzederek sorgu şöyle yazılmaktadır :

```
Select muster_i_Adi, muster_i_sehri
into $a, $b
from mevduat, muster_i
where mevduat.muster_i_adi = muster_i.muster_i_adi and
mevduat.bakiye > $x
```

System R ön işlemcisi için komutlar "\$" işaretiyle başlamaktadır. Ön işlemci dört adet komuta sahiptir. Bunlar \$declare, \$let, \$open, \$fetch komutlarıdır.

¹⁰⁵ Korth and Silberschatz, a.g.e., p. 481-482

Yukarıdaki örnek bu komutlara göre dizayn edilecek olursa :

```

$declare a char (20) ;
$declare b char (20) ;
$declare x binary fixed (15) ;
declare rc binary fixed (15) ;
      $let c be select musteri_Adi, musteri_sehri
      into $a, $b
      from mevduat, musteri
      where mevduat.musteri_adi = musteri.musteri_adi and
      mevduat.bakiye > $x

$open c;
rc = 0 ;
do while rc = 0 ;
      $fetch c;
      put skip list (a,b) ;
end

```

İyileştirme yöneticisi hem gölge sayfalamayı hem de günlük kaydını kullanmaktadır. System R ticari bir ürün olmaktan ziyade bir araştırma projesi olması nedeniyle açık literatürde iyi belgelenmiştir¹⁰⁶.

1.25.1.2. Database 2 (DB2)

DB2, MVS işletim sistemi ile çalışan ticari bir veritabanı sistemidir. IBM tarafından geliştirilmiştir. DB2 ve SQL/DS ayrı sistemler olmalarına rağmen aralarında yüksek düzeyde bir benzerlik vardır. Her ikisi de SQL sorgu dilinin neredeyse aynı versiyonunu ve gömülü SQL için aynı söz dizimini kullanır. Bu iki sistem veriyi doğrudan paylaşamaz fakat aralarında veri transferi mümkündür. DB2 temel kavramları da SQL'e -dolayısıyla System R'ye de- benzerdir. Bu programın ilk özelliği, geniş veritabanlarını yönetecek büyüklük ve kapasiteye sahip olmasıdır. SQL/DS'ten farklı olarak DB2 IMS'ye rakiptir. 1983'te sunulan DB2 ticari çevrelerde ilişkisel veri modelinin artan ve hiyerarşik modellerin azalan önemini gösterir. Daha önce incelenen iki safhalı işlem protokolü DB2'da başarıyla uygulanmıştır. Sorgu yönetim imkânı, veritabanı ile etkileşime izin veren bir DB2 arabirimidir¹⁰⁷.

1.25.1.3. Oracle ¹⁰⁸

Oracle Corporation'ın Oracle veritabanı sistemi, IBM tarafından geliştirilmiş bulunsa da SQL/DS'nin sorgu dili ve kullanıcı boyutuna yakından uymaktadır. 1992 yılında dünya veritabanı pazarında ilk sırayı almıştır. Tüm ana protokolleri desteklemektedir. Tüm donanım ve işletim sistemi platformlarında (Unix, VMS, MVS, VM, HP MPE/XL, SIEMENS, ICL, DEC, VAX, IBM/370, OS/2, Macintosh, Nowell Netware)

¹⁰⁶ Korth and Silberschatz, a.g.e., p. 481-486

¹⁰⁷ Korth and Silberschatz, a.g.e., p. 481-489

¹⁰⁸ Anonymous, BYTE Türkiye, Kasım 1996, Dr. Yalçın Özkan ve Gülser Acar Dondurmacı, Oracle Mimarisi, Cilt : 3, Sayı : 11, İhlas Holding A.Ş., s. 174-177

çalışabilmektedir. Nowell Netware 3.11 üzerinde en az 16 MB bellek gerektirirken, DOS için en az 8 MB Ram ve işlemci olarak da 386SX gerekmektedir.

Oracle, yüksek derecede güvenilir ve kullanıcıdan bağımsız dağıtık bir veritabanına sahiptir. Bu tür sistemlerde kullanıcının kullanması gereken bilgiler farklı yerlerde depolanmış olsa bile tüm bilgilere ulaşılabilir¹⁰⁹. Gömülü SQL sorgu dilini ve bu kullanım için uygulama programlarını mümkün kılar. SQL'in önerildiği ana diller Cobol, Fortran ve C'dir. Gömülü SQL'e ilaveten SQL ifadelerinin çalışma zamanında yorumlandığı bir çağrı arabirimine de sahiptir. Çoğu ticari sistem gibi rapor üretme imkânları bulunmaktadır. İndekslerde anahtar sıkıştırma tekniğini -indeksteki bütün anahtarları depolamaktan ziyade sadece izlenecek doğru işareti tanımaya yetecek bir anahtar parçasını depolama- kullanmaktadır. Bu isim ve adres alanları gibi uzun anahtarlar için oldukça önemli bir tasarruf sağlamaktadır¹¹⁰.

ORACLE, geçtiğimiz Haziran ayının sonlarında ORACLE8'i duyurmuş bulunmaktadır. Bu ürün ilişkisel veritabanı sisteminin kullanım kolaylığını korurken aynı zamanda nesne-temelli bir veritabanı özelliğine de sahiptir.

1.25.1.4. Informix

Informix'in İlişkisel DBMS'i dünya pazarında önemli bir yere sahiptir. Özellikle Unix bazlı veritabanı satışında ilk sıralarda yer almaktadır. Amerikan ordusunun önemli projeleri bu programla yapılmıştır. Şirket dünyada en fazla 4. kuşak dil geliştiren firma durumundadır ve bu pazarın % 50'sinden fazlasına sahiptir. Yüksek miktarda bilgi depolaması gerektiren ve yoğun erişim istenilen alanlarda kullanılmaktadır. Güvenlik kontrolleri kolaylıkla yapılabilen, kolon bazında gizlilik, kullanıcı önceliği işlemleri rahatlıkla yürütebilmektedir. Performans ve hızının fazla olması, dördüncü kuşak dilleri sunması, ürün yelpazesinin genişliği, ses ve grafik işleme yeteneklerinin bulunması önemli üstünlükleridir. Değişik ortamlarda da çalışabilmektedir. Ayrıca uygulamaları basitleştirici CASE tool araçlarına sahiptir. Programın en güçlü özelliği dağıtık veritabanılarını desteklemesi, birçok platformda işlemesi ayrıca maksimum 2GB'lık bloklar halinde resim saklama ve işleme yeteneğine de sahip bulunması gösterilmektedir.

1.25.1.5. Progress

Progress, sayısal ve karakter tabanlı bilgilerin doğrudan saklanabildiği gibi ses ve görüntü gibi çok farklı yapılarıdaki bilgileri ve bu bilgileri içeren dosyaların adreslerini veritabanında saklaması sayesinde geleneksel uygulamalarla entegre olarak kullanılabilir. Veri bütünlüğü, performans, merkezi dosya kayıt ve kural tanımlayabilme imkânları ürüne yüksek bir verimlilik kazandırmaktadır. 15 MB yer kaplamakta olup 2 MB bellek ile verimli bir şekilde kullanılabilir. En az 80386SX'li bir sistem yeterli olmaktadır. Progress aynı zamanda tam anlamıyla bir taşınabilirlik ve ağ desteği sunmaktadır¹¹¹.

¹⁰⁹ Anonymous, Bilgisayar(Datamation) Dergisi, a.g.d., s. 49

¹¹⁰ Korth and Silberschatz, a.g.e., p. 489-490

¹¹¹ Anonymous, Bilgisayar(Datamation) Dergisi, a.g.d., s. 47-48

1.25.2. Şebeke Veritabanları

Şebeke modeli, en eski veritabanı sistemlerinin çoğunun ve bazı yeni sistemlerin ise temelini oluşturmaktadır. Burada iki popüler şebeke sisteminden, Total ve IDMS'den söz edeceğiz.

1.25.2.1. Total

Total veritabanı sisteminin tarihi 1960'ların sonlarına kadar uzanmaktadır. Minikompüterlerden Mainframe'lere kadar geniş yelpazede makinalarda çalışmak üzere tasarlanmıştır. DBTG modeline dayanmasına rağmen sorgu dili farklıdır. Cobol, PL/1, Fortran ve RPG içinden kullanılabilir.

1.25.2.2. Idms

Cullinane Database Systems Inc. tarafından geliştirilen bir şebeke veritabanı sistemidir. DBTG modeline yakından bağlıdır ve veritabanı tasarımcısına veritabanının fiziksel organizasyonu üzerinde yüksek düzeyde kontrol imkânı sunan ayrıntılı veri manüpilyasyon dili içermektedir. Artan programcı kontrolü ile daha etkin sorgular yazılmasını mümkün kılacak ilave nitelikler eklenmiştir.

1.25.3. Hiyerarşik Sistemler

Hiyerarşik model IBM'in IMS veritabanı sisteminden dolayı öncelikli bir öneme sahiptir. Bu bölümde hem IMS, hem de bir başka yaygın kullanıma sahip System 2000 incelenecektir.

1.25.3.1. IMS (Information Management System)

IBM Bilgi Yönetim Sistemi (IMS) en eski ve en yaygın veritabanı sistemlerinden biridir. IMS veritabanları, tarihsel olarak en geniş veritabanları arasında sayıldığından, IMS geliştiricileri eşzamanlılık (concurrency), onarım (recovery), bütünlük (integrity) ve etkin sorgu işleme gibi konularla ilgilenmek zorunda kalanların yanısında yer almışlardır. Hayli kompleks bir yapısı vardır. Gömülü çağrılar, IMS veritabanı dili olan DL/1'in parçasıdır. Geniş veritabanlarında, performans hayati bir önem taşıdığından veritabanı tasarımcısına, veri tanımlama dilinde çok sayıda seçenek sunar. IMS de çeşitli kayıt erişim şemaları mevcuttur¹¹².

1-HSAM (Hiyerarşik Sıralı Erişim Metodu) : Fiziksel olarak -tape dosyaları gibi- ardışık dosyalar için kullanılır. Kayıtlar fiziksel olarak sırayla depolanır.

2-HİSAM (Hiyerarşik İndeks Sıralı Erişim Medodu) : Hiyerarşinin en alt düzeyinde (root) bir indeks sıralı düzenlemedir.

3-HİDAM (Hiyerarşik İndekslenmiş Doğrudan Erişim Metodu) : Yeni (child) kayıtlar için en alt düzeyde yapılan indeks düzenlemeleridir.

4-HDAM (Hiyerarşik Doğrudan Erişim Metodu) : HİDAM'a benzetmekle beraber en alt seviyede bir erişim bulunmamaktadır.

¹¹² Korth and Silberschatz, a.g.e., p. 492-493

IMS'nin orijinal versiyonu eşzamanlı kontrol teorisinin gelişiminden önce ortaya çıkmıştır. Bu uyarılama, bir kerede sadece bir güncelleştirme uygulaması programı işleyebilmekteydi. Daha sonraki versiyonları hem geliştirilmiş eş zamanlı kontrolü hem de (log gibi) daha karmaşık işlem onarım tekniklerini kullanan karmaşık program izolasyon özelliği içermektedir. Başlangıçta kural olan bBatch işlemlere karşın, IMS kullanıcıları on-line işlemlerini kullanmaya başladıkça bu özelliklerin önemi artmıştır. IMS'nin Fast - Path özelliği, veritabanının en aktif parçalarının ana hafızada kalmasını mümkün kılacak şekilde dizayn edilmiştir. Veritabanı, SQL kullanıcıları tarafından güncelleştirilememesine rağmen sorgulanabilmektedir.

1.25.3.2. System 2000

System 2000, İlk olarak MRI Corporation tarafından geliştirilen, şimdilerde Intel'in yürüttüğü hiyerarşik bir veritabanı sistemidir. IBM 370 ve benzeri, Univac 1100 v.b. bilgisayarları üzerinde kullanılabilir. Veri modeli IMS ile aynı olmasına rağmen sunduğu dil özellikleri, IMS DL/1 dilinden birkaç yönden ayrılmaktadır. SYSTEM 2000 sorgularını bir ana dile yüklemek mümkün olup, program ön derleyicisi bu programı işleyip, gramer açısından kontrol etmektedir. Sorgu dilinin adı "Quest" olarak bilinir.

1.25.4. Mikrobilgisayar Veritabanı Sistemleri

Bu veritabanı sistemlerinin özellikleri basit olmalarıdır. Kişisel bilgisayarların sınırlı gücü, hem veritabanı büyüklüğünü hem de sistemin karmaşıklık düzeyini sınırlandırmaktadır. Bu sistemlerde her bir kullanıcı aynı zamanda veritabanı yöneticisi olarak hizmet etmektedir. Büyük sistemlerle mukayeseleri ise aşağıda verilmiştir.

- 1-Veri modeli : Mikrobilgisayar veritabanı sistemleri, yeni olduklarından hemen hemen hepsi ilişkisel modele dayanır.
- 2-Sorgu dili : Büyük sistemlerin veritabanı ile olan etkileşimi genellikle bir form arabirimine dayanır.
- 3-Fiziksel uygulama : Nesne kodunun işgal ettiği yer sistem uygulayıcıları için önemli bir konudur. Yer ihtiyacını azaltarak sistemi daha az ana hafızaya sahip makinalarda çalıştırmak mümkündür. Bu potansiyel piyasa üzerinde büyük bir etkiye sahip olabilir.
- 4-Onarım : Çoğu sistem onarım altsistemine sahip değildir. Kullanıcının veriyi düzenli olarak yedeklemesi beklenmektedir.
- 5-Eşzamanlılık : Bu kontrole tek kullanıcı kişisel bilgisayarlarda ihtiyaç yoktur.

Günümüzde, mikrobilgisayar veritabanı sistemleriyle, büyük sistemlere ait veritabanı sistemleri arasındaki farklılıklar giderek azalmaktadır¹¹³.

1.25.4.1. dBASE-III

Ashton-Tate'nin dBASE-III'ü ticari olarak başarılı dBASE-II ürününün bir versiyonudur. Hem bir veri manipülasyon dili hemde genel amaçlı bir programlama dilini kapsamaktadır. Bu programla doğrudan veritabanına erişim sağlanabilmektedir. Sorgu dilini gücü oldukça sınırlıdır. Bu programdan sonra Borland firması 1993 yılında dBASE

¹¹³ Korth and Silberschatz, a.g.e., p. 494-497

III'e göre yetenekleri bir hayli gelişmiş bir veritabanı programlama dili olan dBASE IV'ü¹¹⁴ piyasaya sürmüştür.

1.25.5. Spreadsheet Veritabanı Sistemleri

Software Arts firmasının bir ürünü olan Visicalc ilk başarılı Spreadsheet diliydi. O kadar ki rakipleri "PC'nizde Lotus 1-2-3 dışında programlar da çalışır" diye reklamlar vermeye başlamışlardı. Pratik ve veritabanı özellikleriyle güçlendirilmiş elektronik bir çalışma sayfasıdır. Spreadsheet kelimesi, Türkçe'ye hesaplama tablosu veya hesaplama sayfası olarak tercüme edilmektedir. Geleneksel programlama dillerinden farklı olarak iki boyutlu hücreler düzleminden oluşmaktadırlar. Bu sistemlerle programlama, hücreler arası matematiksel ilişkilerin tanımını içerir veri girildikçe bu ilişkiler sistem tarafından devam ettirilir. Her bir hücre kendisiyle ilişkili bir değer formülüne sahip olmaktadır. Bu tür yazılımlara örnek olarak Excel, Quattro Pro, Wingz sayılabilir.

1.25.6. Client/Server Veritabanı Sistemleri

Bir hayli popüler olan Client/Server mimarisinde, merkezi bilgisayar, veritabanının tüm yönetiminden ve diğer kullanıcıların erişimlerinden sorumludur. İş istasyonu sıfatına sahip bilgisayar, tüm giriş-çıkış işlemlerini üzerine alır ve Server'i bu yükten kurtararak grafiksel kullanıcı arabiriminin (GUI-Grafik User Interface) verimli bir şekilde kullanılmasına imkân verir. Amaç server ve ona bağlı olan iş istasyonu arasında bir denge tesis etmektir.

Ağ üzerinde kurulu bir veritabanı sistemindeki verilerin önemi, geniş alan ağlarında (WAN) ortaya çıkmaktadır. Merkezi veri yığınının işlenmesi depolandığı yerde olmakta, yani veritabanı sistemi yerel iş istasyonunda değil, yerel ağın Server'i üzerinde çalışmaktadır. Bilgi işlem alanında kullanılan donanım açısından Client/Server, merkezi host veya mainframe sistemlerinden dönüş olarak görünmektedir. Bir kullanıcının çalıştığı veritabanı uygulaması (front end) kullanıcının yerel bilgisayarında çalışır. Buna Client adı verilir. Yalnızca yeni verilere ihtiyaç duyulursa, ağ üzerinden kodlanmış olarak gelen bilgi transferi gerçekleşmekte, veritabanı sistemi (back end) bu kodlu bilginin yardımı ile sadece ihtiyaç duyulan bilgiyi arayıp ağ üzerine bulduğu verileri Client'a göndermektedir. Sonuç olarak kaynakların dengeli kullanımı, ağ üzerindeki azaltılan trafik ve yüksek işlem hızı elde edilmektedir.

1.25.6.1. Ingres

IBM San Jose Research Laboratuvarı'nın, Sistem R prototipini geliştirmesiyle hemen hemen aynı zamanda Berkeley University of California'da bir grup, Ingres adı verilen deneysel bir veritabanı sistemini geliştirdiler. İlişkisel veri modeline dayanmalarına rağmen sistem dizaynı ve kullanıcı boyutunda önemli farklılıkları bulunmaktadır. Bu araştırma projesi aynı adla anılan ve Relational Technology tarafından geliştirilen ticari bir ürün oldu¹¹⁵. 1992 yılında yurtdışı veritabanı piyasasında Oracle'dan sonra ikinci sırada yer almıştır. Client/Server mimarisi ile çalışmakta olan Ingres, multi-user ve multi-

¹¹⁴ Bkz. Atilla Büke, 1991, DBase IV El Kitabı, A&S Yayın Tanıtım, Ankara.

¹¹⁵ Korth and Silberschatz, a.g.e., p. 490-501

tasking ortamlarda çalışabilmektedir. Ingres'in ana yapısını oluşturan veritabanları, Server ve Client'ler farklı ortamlarda bulunabilmektedir¹¹⁶.

Ingres ilk olarak Unix işletim sistemini kullanan PDP-11 bilgisayarı üzerine geliştirilmiştir. Unix, Ingres'in dahili yapılanma biçimini oldukça etkilemiştir. Herbiri özel bir amaçtan sorumlu birkaç süreç (process) kullanılmıştır. Bu süreçler :

- 1- Sorgu formülasyonu, kullanıcı etkileşimi
- 2- Kelime analizi ve gramer inceleme, sorgu optimizasyonu
- 3- İndeks hizmetlerinin yürütülmesi

Bir kullanıcı isteği, gerektiği zaman bir süreçten ötekine iletilmektedir. Bu süreç çokluğu sayesinde Ingres, kendi içinde eşzamanlılığı sağlamaktadır. Tarihsel olarak Ingres dizaynını motive eden temel faktör PDP-11'in her bir süreç için 64 Kb'lık adres alanı sınırlaması olmuştur. Ingres akabinde, QUEL sorgu dilini piyasaya sunmuş ve QUEL'in C ana diline yüklenmesini EQUQL olarak adlandırmıştır. "###" karakterleri bir QUEL ifadesini C programında göstermek için kullanılmaktadır. EQUQL, QUEL sorgusu sonucunda her bir satır için belli kod yapısının bir kez icra edildiği daha yapısal bir yaklaşım kullanmaktadır.

```
## char a[20], b[20] ;
## float x ;
    main () {
    ...
    ...
    ...
## range of d is depozit
## range of d is musteri
## retrieve a = c.musteri_Adi, b= c.musteri_sehri
## where d.bakiye > x
#{printf ("%5, %7.2f\n", a,b,)
#}
    ...
```

Yukarıda printf komutunda görülen "#{ " ve "#}" işaretler sonucun her bir satır için bir kez icra edilmesini sağlamaktadır¹¹⁷. Ingres veritabanında yer alan bilgiler, klasik veriler (karakter, numerik, zaman v.b.) olduğu gibi bilgi ve kurallar (Kasa hesabı hiçbir zaman alacaklı olamaz ya da endeks belli bir oranın altına düşerse hisse alınmalı gibi) ve nesneler (resimler, koordinatlar, eğitim, derece gibi) de olabilmektedir. Diğer önemli ilişkisel DBMS'lerden farkı beraberinde Bilgi ve Nesne Yönetimini de içermesidir. Ayrıca taşınabilir DBMS'lerde ilk defa iki safhalı kabul protokolü işlemini gerçekleştiren Ingres, dönüşümsel yapıdaki log dosyaları ve arşivleyici fonksiyonları sayesinde tam bir veri güvenliği sağlamaktadır.

¹¹⁶ Anonymous, Bilgisayar(Datamation) Dergisi, a.g.d., s. 46

¹¹⁷ Korth and Silberschatz, a.g.e., p. 491-492

Diğer ilişkisel DBMS'lere (DB2, IMS v.b.) erişim amacıyla kullanılan ve adı Ingres Gateway olan versiyonu, yabancı bir veritabanına ait verilerin yerel Ingres verileri olarak kullanılmasını sağlamaktadır. Ingres Star ise birden çok veritabanına, yerleşimden bağımsız olarak erişimi sağlamaktadır¹¹⁸.

1.25.7. Windows Veritabanı Sistemleri ¹¹⁹

Günümüzde çeşitli arabirimler sayesinde veritabanı çalışmaları oldukça kolaylaşmıştır. Derin bir veritabanı bilgisi gerektirmeksizin en zor sorgulamalar bile menüler aracılığı ile yapılabilmektedir. Bu kolaylık uygulamaların basitce oluşturulmasını da sağlamaktadır.

Delphi, Borland'ın derleyici teknolojisindeki en iyi yanlarını alan, bunları Borland'ın veritabanı teknolojisindeki en iyi yönleriyle birleştiren, yeni görsel programlama araçlarını kullanan özgün bir üründür. Veritabanı işlemlerini kısa zamanda yerine getirmek için Borland Veritabanı Motorunu kullanmaktadır. Veritabanı programcıları iki tablo arasında bire-çok ilişkileri gösteren formları bile otomatik olarak oluşturabilmektedirler. Karmaşık SQL deyimleri oluşturmaya yarayan görsel araçlar içermekte, editörü olan Dataset kullanılarak kullanıcıların tablo alanlarına erişimleri denetlenebilmekte, verinin kullanıcıya nasıl gösterileceği, kullanıcı veri girerken hangi tuşları kullanacağı dahi belirlenebilmektedir. Tam olarak ölçeklenebilir bir veritabanı aracıdır. Bu şu anlama gelmektedir. Tek kullanıcı bir makina için yazdığınız kodu sadece birkaç küçük değişiklik yaparak ağ üzerinde çalışabilecek hale getirmek mümkün olmaktadır. Bunu sağlayan araç Local Interbase Engine'dir. Interbase belli başlı kurumlar tarafından kritik görevlerde kullanılan bir SQL sunucudur. Bu ürün aynı zamanda Paradox veya dBASE dosyalarını kullanan tüm işlemlere sahip bir DBMS dir¹²⁰.

Windows altında çalışabilen diğer veritabanları ise Access 2.0, Paradox 5.0, dBASE 5.0, Sqlbase, Oracle7 Release 7.2, Visual Foxpro v.b. sayılabilir.

¹¹⁸ Anonymous, Bilgisayar(Datamation) Dergisi, a.g.d., s. 46

¹¹⁹ Anonymous, Nisan 1996, PC World Türkiye, Windows Veritabanları, Sayı 31, IDG/UFT Yayıncılık, s. 78-86

¹²⁰ Charles Calvert, Çev. Hasan F. Keçeci ve Esra Atalay, Şubat 1997, Delphi Unleashed, Sistem Yayıncılık, İstanbul, ISBN : 975-322-010-3, s. 10

2. KÜTÜPHANE UYGULAMASI

2.1. Kütüphaneler

Kütüphanelerin ekonomik, teknik, sosyal ve kültürel kalkınmadaki önemi inkar edilemez bir gerçektir. Amaçları, her türden fikir ve sanat ürünlerini bünyesinde barındırıp kullanıcılarına dökümantasyon ve enformasyon hizmeti vermek, bu yolla kültür birikimi oluşturmaktır. Bu amaçlarına ulaşmak için ellerindeki malzeme ve bilgileri hem okurlarıyla hem de birbirleri ile paylaşmakta, çeşitli kataloglar yayınlarak bilginin kolayca erişilebilirliğini sağlamaktadırlar. Günümüzde kütüphaneler bilgisayar ağlarını kullanarak bilgi alışverişini daha kolay ve hızlı bir şekilde yapmaktadırlar. Kütüphanelerde kitapların yanı sıra gazeteler, dergiler, haritalar, afişler, şemalar, taş plaklar, bantlar ve bunları dinlemeye yarayan araçlar, özel çalışma alanları, konferans salonları, stüdyolar bile bulunabilmektedir. Hatta özürli kişileri de kütüphanelerden yararlandırmak için çalışmalar yapılmaktadır.¹²¹

2.2. Kütüphanecilik Hizmetleri

Kütüphanelerin verdiği hizmetler genel olarak 3 ana başlık altında ele alınmaktadır. Bunlar sağlama (aksesyon), hazırlama ve yararlandırma olup kütüphanecilik ise bu hizmetleri kendine konu edinen ilim dalının adıdır. Kitaplar hakkında bilgi veren ve onları çeşitli yönlerden tanıtan ilim dalına ise Bibliyografya denilmektedir.

2.2.1. Sağlama

Kütüphanelere kitap, süreli yayın ve diğer araçlar üç yolla sağlanmaktadır. Satın alma, bağış ve değişim. Ayrıca derleme adı verilen bir yöntemle de kitaplar temin edilmektedir.

2.2.2. Hazırlama

Çeşitli yollarla kütüphaneye giren kitap v.b. dökümanlar yararlandırma aşamasına kadar sınıflandırma, kataloglama, ciltleme ve yerleştirme gibi çeşitli teknik işlemlerden geçmektedir.

Sınıflandırma (tasnif) işlemi, kitapların ihtiva ettikleri konulara göre sınıflandırılmasıdır. Bunun için dünyada bugüne kadar çeşitli sistemler kullanılmıştır. Brunet, Schleiermacher, Hartwig, Wieser, Dwey v.b. Bunlar arasında en yaygın olarak ülkemizde de benimsenmiş olanı, 1876'da Amerikalı kütüphaneci Melvil Dewey (1851-1931) tarafından ortaya konan ve kısaca Dewey Sistemi veya Dewey Onlu Tasnif Sistemi denen Decimal Classification and Relativ Index'dir. Bu sistemde konular 0'dan 9'a kadar on ana sınıfa ayrılmıştır. Bu on sınıftan her biri de kendi arasında 10'arlı tali gruba ayrılmıştır. Bu tali grupların her birini de tekrar 10'arlı gruba ayırmak mümkündür. Böylece bu sistemde, kütüphanenin büyüklüğüne ve kitap adedine göre istenildiği kadar detaya inilebilmekte ve her bir konu ayrıntılı bir biçimde değişik tasnif numaraları ile gösterilebilmektedir. Dewey Sistemi'nde 10 ana grubun ifade ettiği konular Şekil 1'de gösterilmiştir.¹²²

¹²¹ Bkz. Sten Gustavs, çev. Neslihan Uraz, Mart 1996, İsveç'te konuşan kitaplar-Görme Özürlüler İçin gelişmekte olan Bir Olanak, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 10, Sayı 1, ISSN : 1300-0039, s. 44-47

¹²² Anonymous, Dergâh Yayınları, Türk Dili ve Edebiyatı Ansiklopedisi, 1986, İstanbul, 6. Cilt, s. 52

Tablo 4. Dewey Onlu Tasnif Sistemi

Konu No	Konu Adı
000	Genel Konular
100	Felsefe
200	Din
300	Sosyal İlimler
400	Dilbilim
500	Nazarî İlimler
600	Uygulamalı İlimler
700	Güzel Sanatlar
800	Edebiyat
900	Tarih, Coğrafya, Hal tercümesi

Kataloglama işleminde amaç kitap, dergi, gazete v.b. araçların belli kurallara göre künyelerinin tesbit edilmesidir. Kataloglar, kitap ve fiş katalogu olmak üzere iki çeşittir. Kitap (cilt) katalogu, isminden de anlaşılacağı üzere alfabetik veya sistematik bir düzen içerisinde kitap şeklinde basılmış eserlerdir. Bunlara fihrist de denmektedir. Fiş katalogu, sisteminde ise, kütüphanede bulunan kitap, dergi, gazete v.b. araçların künyeleri 7.5 x 12.5 cm. boyutlarındaki standart karton fişler üzerine tesbit edilerek fiş toplulukları oluşturulmaktadır. Fiş katalogları, alfabetik ve sistematik olmak üzere iki şekilde dizayn edilmektedir. Alfabetik katalogda fişler, yazar soyadı, adı ve kitap adına göre alfabetik düzen içerisinde hazırlanırken, sistematik katalogların esasını kitapların ana ve alt ilim dallarına göre aldıkları konu numaraları teşkil etmektedir.

Kataloglama yapılırken genel olarak her kitap için yazar, kitap adı ve konu fişi çıkarılmakta ve yazar fişi esas fiş olduğundan önce bu fiş hazırlanmaktadır. Diğer fişler yazar fişine göre çıkarılmaktadır. Fiş katalogunun asıl amacı aranan kitabı çeşitli yollardan kolaylıkla bulmayı sağlamaktır. Bu sebeple yukarıda adı geçen üç fişin dışında gerekirse ikinci ve üçüncü yazar, çeviren, anlatan, derleyen, toplayan, hazırlayan, yayınlayan, tamamlayıcı kitap v.b. bilgiler için "ek fiş"; takma adlar, Soyadı Kanunu'ndan önceki isimler ve kadınlarda farklı soyadları için ise "işaret fişi" çıkartılır. Fiş kataloglarına kitabın künyesi aşağıdaki sıra dahilinde geçirilmektedir¹²³.

- 1-Kitabın yer numarası
- 2-Konu numarası
- 3-Yazar adı
- 4-Kitap adı ve tamamlayıcı ad
- 5-Cilt kaydı (Kaç cilt veya kaçınıcı cilt olduğu)
- 6-Basım kaydı (Kaçınıcı basım olduğu)
- 7-Çeviren, derleyen, toplayan, hazırlayan, yayınlayan v.b.
- 8-Baskı kaydı (Nerede ne zaman basıldığı, yayınevi, kitabevi veya matbaa)
- 9-Sayfa kaydı

¹²³ Dergâh Yayınları, a.g.e., s. 52

10-Resim, plan, kroki, harita, cetvel v.b.

11-Çok ciltli eserlerde muhteva kaydı

12-Seri kaydı ve not

Ciltleme aşamasında, kütüphaneye değişik yollardan giren kitaplardan ciltlessiz olanları önem sırasına göre ciltleme işlemine tabi tutulmaktadır. Müracaat kitapları, gezici kütüphane kitapları, çok kullanılan kitap ve süreli yayınlar, kapaksız, dikişsiz ve formasız kitaplar, cilt bütünlüğü kazanmış ve yeni sayıları gelmeye başlamış olan kitapların öncelik sırası vardır.

Yerleştirme safhasında ise yukarıdaki teknik işlemler tamamlandıktan sonra kitaplar etiketlenir ve depo veya salondaki yerine yerleştirilir. Günümüzde daha çok yayınların kütüphaneye geliş sırası veya konu esaslı yerleştirme sistemleri kullanılmaktadır. Kitap mevcudu çok olan kütüphaneler için geliş sıralı, az olanlar için konularına göre yerleştirme en ideali olmaktadır.

2.2.3. Yararlandırma

Kütüphanelerin asıl gayeleri yararlandırmadır. Sağlama, hazırlama v.b. hizmetler bu amaca ulaşmak için sadece birer araçtır. Kütüphaneden yararlanma oranının büyüklüğü kütüphanenin o nisbette görevini yaptığını göstermektedir. Kütüphanelerden iki şekilde yararlanılmaktadır. Biri kütüphane araçlarından kütüphane içerisinde faydalanmak, diğeri bu araçları ödünç alarak kütüphane dışında yararlanmak. Kütüphaneler, türlerine ve amaçlarına göre bu hizmetlerden bir veya ikisini verebilmektedirler. Bu hizmeti verirken göz önünde tutulması gereken nokta, istenilen her türde dökümanı en kısa zamanda kullanıcıya ulaştırabilmek için gerekli tedbirlerin almak olacaktır.

2.3. Bilgi, Kütüphane ve Kütüphane Otomasyonunun Önemi ¹²⁴

Kütüphane kurumu ve kütüphane hizmetlerinin hammaddesi bilgidir. Günümüz bilgi çağını yaşamakta ve hızla bilgi toplumu olma yolundadır. Bilginin önemini maddeler halinde sıralayacak olursak :

- 1- Kalkınma ve GSMH'nın gelişimi için enerji kadar önemli ve değerli bir kaynaktır.
- 2- Sosyal refah ve toplum barışı için hayati değer taşımaktadır.
- 3- Uluslararası ilişkilerdeki dengelerin korunması için zorunludur.
- 4- Her türlü üretim safhasında (örneğin; karar üretimi, program üretimi, araştırma üretimi, mal ve ürün üretimi, hizmet üretimi v.b.) vazgeçilmez temel girdilerden birisidir.
- 5- Toplumun her kesiminde, bilgi ihtiyacı giderek belirgin ve yaygın olarak hissedilmekte ve hissettirilmektedir.
- 6- Karar alma sürecinin her kademesinde ve her düzeyindeki görevli, planlamacı, yönetici, araştırmacı, bilim adamı, eğitimci, uygulayıcı, öğrenci, her türlü bireysel, toplumsal ve ekonomik faaliyetlerini etkin ve sağlıklı bir biçimde gerçekleştirebilmek için, dünün ve bugünün bilgi birikiminden yeterince, doğru olarak ve hızlı bir biçimde yararlanmak ve yararlandırılmak zorundadır.

¹²⁴ Anonymous, 1985, Türkiye'de Kütüphanecilik Alanında Bilgisayar Uygulamaları 1. Sempozyumu Sonuç Raporu, İstanbul Üniversitesi Edebiyat Fakültesi Kütüphanecilik Bölümü, İstanbul, 1985 - AD - 4380, s. 3

Bu denli önem taşıyan bilgi, ancak kullanıldığı ve yararlanıldığı ölçüde değer kazanmaktadır. Bilginin en yüksek düzeyde kullanımını ve yararlanılmasını sağlayan kurum kütüphanedir. (Kütüphane terimi; bilgi ve belge merkezi, arşiv ve her türlü bilgilendirme kurumlarını kapsayacak anlamda kullanılmıştır). Kütüphane, bilgi iletişim sürecinin en önemli halkalarından birisidir. Bilginin kaynağı ile kullanıcı arasında bağlantıyı sağlar, bilgiyi kullanıcıya ulaştırır. Kütüphane, dünyanın neresinde üretilirse üretilsin bilginin denetim altına alınması yani, varlığının belirlenmesi, toplanması, düzenlenmesi ve her kesim ve düzeydeki bilgi kullanıcısına en hızlı, en uygun ve doğru bir biçimde ulaştırılmasını amaçlayan bir kurumdur. Bu amacını gerçekleştirebilmek için, kütüphane, bir dizi teknik hizmetler gerçekleştirir ve çeşitli kütüphane hizmetleri ya da bilgi hizmetleri sunar.

2.4. Kütüphane Otomasyonunun Amacı ve Önemi ¹²⁵

Kütüphane kurumu, amaçlarını en verimli ve rasyonel olarak gerçekleştirebilmek için modern teknolojinin sağladığı imkânlardan yararlanmak zorundadır. Bu teknolojik araçlardan birisi olan bilgisayar, bütün diğer kurumlara olduğu gibi, kütüphane kurumuna da geniş imkânlar sağlamakta, her türlü kütüphane hizmetinin etkin ve hızlı olarak yürütülmesinde başarıyla kullanılmaktadır. Günümüz dünyasında, bilgilenme ve bilgilendirme hususunda bilgisayar kullanımı yani Bilgi Otomasyonu ya da Kütüphane Otomasyonu tek seçenek haline gelmiştir¹²⁶.

Kütüphane Otomasyonu ile toplumun her kesimindeki bilgi kullanıcısının bilgi ihtiyacını karşılamak, dünyanın neresinde üretilirse üretilsin bilgi birikiminden yararlanmak, ülkenin bilgi kaynaklarını ve bilgi potansiyelini belirlemek ve kullanıma sunmak, toplumda bilgiye karşı talep meydana getirmek, bilgi edinme ve bilgi kullanma alışkanlığını geliştirmek amaçlanmaktadır. Ayrıca bilgi toplumu bilinci oluşturmak için çok önemli bir araç ve bir mecburiyettir. Kütüphane otomasyonuna gidilmesinin sebeplerini şöyle sıralayabiliriz¹²⁷.

- 1- Kütüphanede bulundurulması gereken bilgi ve belge miktarı çok büyük boyutlara ulaşmıştır.
- 2- Bilgi içeren her türlü belge (kitap, makale, dergi, rapor, tez, patent, tebliğ, yasa, gazete, harita, fotoğraf, bant, slayt v.b.) hem biçim olarak, hem konu olarak çeşitlilik kazanmış ve artmıştır.
- 3- Kütüphanelerde:
 - a- Çok yoğun emek gerektiren işler vardır.
 - b- Çok fazla rutin iş vardır.
 - c- Yapılan işlerde çok fazla tekrar vardır. Örneğin, aynı bilgileri içeren kayıtlar tekrar üretilmekte, aynı bilgi değişik amaçlar için tekrar tekrar kullanılmaktadır.
 - d- Bazı temel işlemler her kütüphanede aynen tekrarlanmaktadır.
 - e- Kütüphanede üretilen kayıtlar, değişik amaçlar için çok sayıda çoğaltılmaktadır.
- 4- Toplumda bilgi bilinci artmakta ve bilgi ihtiyacı dakikleşmektedir.

¹²⁵ Anonymous, Türkiye’de Kütüphanecilik Alanında....., a.g.e., s. 4

¹²⁶ Bkz. Nazan Özenç Uçak, Mart 1995, Kütüphaneleri Etkileyen Teknolojik Yenilikler, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 9, Sayı 1, ISSN : 1300-0039, s. 49-55

¹²⁷ Anonymous, Türkiye’de Kütüphanecilik Alanında....., a.g.e., s. 5

5- Kütüphaneci ve bilgi görevlisi, bilgi kullanımının giderek yoğunlaşan ve dakikleşen bilgi ihtiyacını anında ve doğru olarak karşılayabilmek için, çok fazla vaktini alan rutin fakat temel nitelikteki bazı işlemleri ihmal etme meselesi ile karşı karşıyadır. Problemin çözümü, bilgisayar kullanımı ile gerçekleştirilebilmektedir.

6- Bilgisayar teknolojisinin sağladığı hız, kolaylık ve doğruluk, kütüphane hizmetlerinde ya da daha genel bir deyişle bilgi akımının gerçekleştirilmesinde sağladığı üstünlük, dünyanın gelişmiş ve gelişmekte olan ülkelerinde ve uluslararası kuruluşlarda kabul edilmiş ve kütüphane hizmetlerinde bilgisayar kullanımı, yani Kütüphane Otomasyonu yaygınlaşmıştır.

7- Kütüphane Otomasyonu, bilgi kaynaklarının daha akıllıca kullanımını sağlayan ve savurganlığı önleyen etkin bir yol olarak benimsenmiştir.

8- Bu arada, teknolojinin maliyeti düşmekte ve her büyüklükteki kütüphanede her türlü işlem için uygun donanım gerçekleştirilmiş bulunmaktadır.

Sonuç olarak, Türkiye'de Kütüphane Otomasyonu kaçınılmaz bir mecburiyet durumuna gelmiştir. Bilgisayar teknolojisi ve kütüphane otomasyonunun sağladığı faydalara bir göz atacak olursak¹²⁸ :

1- Kütüphaneler, ilk kuruluş günlerinden bu yana, temel amaçları olan bilgilendirme ve eğitime faaliyetlerini bilgisayar teknolojisi sayesinde daha etkin daha hızlı ve doğru olarak gerçekleştirme imkânına kavuşmuşlardır.

2- Bilgisayar teknolojisi; büyük bilgisayarlar, mini bilgisayarlar, süper miniler ve mikroları ile her büyüklükteki kütüphanede ve her türlü kütüphane hizmeti için etkin ve ekonomik kullanım sağlamaktadır.

3- Bilgisayar teknolojisi sayesinde kütüphaneciler, tekrar gerektiren rutin işlerine asgari düzeyde zaman ayırma ve kullanıcıların ihtiyaçları ile daha yakından ve daha çok ilgilenme imkânına kavuşmaktadırlar.

4- Her türlü bilginin makine tarafından okunabilir (machine readable) ortama yüklenmesi sayesinde, en küçük kütüphaneler bile diğer kütüphanelerin kolleksiyonlarından haberdar olmakta ve yararlanabilmektedirler. Bilgi ve belgenin varlığı, artık içinde bulunduğu kütüphane binasının sınırlarını aşmakta ve bütün kütüphaneciler ve kullanıcıların elleri altına ulaşmaktadır.

5- Kütüphane Otomasyonu sonucu üretilen makine tarafından okunabilir kayıtlar sayesinde bilgi; kütüphaneciler, kütüphaneler, okuyucular, kentler ve ülkeler arasında serbestçe ve hızla dolaşır bir nitelik kazanmaktadır. Katalog kayıtları ve belgelerin statikliği giderilmekte ve yerine dinamizm hakim olmaktadır.

6- Aranan bilginin ya da belgenin hangi ülkenin, hangi kentindeki hangi kütüphanede bulunduğunu anında ortaya çıkaran ve bilgiyi, ihtiyaç duyan kullanıcıya büyük bir hızla ileten bilgisayar ve iletişim teknolojisi sayesinde, bilginin cihanşumül özelliği daha bir önem ve anlam kazanmaktadır.

7- Bilgi kaynaklarının sadece bulundukları yörede ve hatta bulundukları binada kullanım zorunluluğunu ortadan kaldırmaktadır.

8- Kütüphane Otomasyonu, kullanıcı tatminini arttıran ve böylece bilgi kullanımını yaygınlaştıran ve dolayısıyla bilgi üretimini hızlandıran bir itici güç niteliği taşımaktadır.

¹²⁸ Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 6

9- Teknolojinin gereği standardlaşma, kaynak paylaşımı, kütüphaneler ve kuruluşlar arası iletişim ve işbirliğini geliştirmekte, bunun sonucu yöresel, kentsel, bölgesel, ulusal, uluslararası kütüphane ve bilgi ağları geliştirilebilmekte pek çok kütüphaneden oluşan bu ağlar, tek bir kütüphanenin sağladığından daha fazla hizmet vermektedirler.

2.5. Kütüphane Otomasyonunun Uygulama Alanları ¹²⁹

Bilgi taşıyan her türlü materyalin üretiminden başlayarak, ilgili materyalin izlenmesi, toplanması, düzenlenmesi ve bu materyale ya da içindeki bilgiye ihtiyaç duyan kullanıcıya sunulmasına kadar geniş bir alanı kapsayan her türlü kütüphane hizmetinde bilgisayar teknolojisi başarıyla uygulanmaktadır. Bir kütüphanede yürütülmesi zorunlu olan her türlü teknik hizmetler ile ilgili sunulan hizmetler kütüphane otomasyonu kapsamına girmiş bulunmaktadır. Kütüphane Otomasyonunun uygulama alanlarını örneklemek bakımından, teknik olmayan terimlerle, şunları sıralamak mümkündür :

- 1- Bilgi ve/veya belgenin var olup olmadığını izleme
- 2- İstenen belgenin toplanması
- 3- Sipariş (abone ve/veya satın alma) kütüklerinin oluşturulması, siparişin izlenmesi
- 4- Belgelerin bibliyografik ve içerik tanımlarının kataloglama, sınıflandırma, indeksleme, özet çıkarma, katalog kayıtlarının hazırlanması
- 5- Katalog ve sınıflandırma kayıtlarının çoğaltılması
- 6- Kütüphane materyalinin fiziksel olarak kullanıma hazırlanması
- 7- Ödünç verme ile ilgili her türlü işlemin yürütülmesi
- 8- Materyalin fotokopisi ya da mikrokopisini isteme ve bunların sağlanması
- 9- Materyalin saklanması ve korunması
- 10- Hangi kütüphanede hangi belgelerin (kitap, dergi, makale, tez, rapor, patent v.b.) bulunduğunu belirten toplu katalogların oluşturulması
- 11- Her türlü yönetim fonksiyonunun yapılabilmesi ve bilgiye erişim

Bir ülkede gerçek anlamda Kütüphane Otomasyonundan sözdebilmek için İnsan-Makine diyaloguna dayanan bilgi erişim uygulamasının gerçekleştirilmesi gerekmektedir.

2.6. Kütüphane Otomasyonu Alanındaki Gelişmeler ¹³⁰

Kütüphane Otomasyonu alanında dünyadaki gelişmeleri, ayrıntıları ile anlatmak mümkün değildir. Ayrıntıya girmeksizin bu alanda dünyada neler olup bittiği kısaca aşağıda özetlenmiştir.

Yaygın olarak bütün dünya kütüphanelerine, kütüphanelere ve bilgi kullanıcılarına ulaşan ilk Kütüphane Otomasyonu uygulaması, Amerikan Ulusal Tıp Kitaplığı tarafından hazırlanan INDEX MEDICUS dergisinin 1964 yılında, bilgisayar çıktısı olarak basılmasıdır. Bu uygulama sonunda elde edilen, sadece derginin bilgisayar çıktısı şeklinde hazırlanması değildir. Aynı zamanda, dergide yer alan kayıtların bilgisayar ortamına yüklenmiş olması sayesinde bu depodaki bilgilerden yine makine ile yararlanma ve bilgiye erişim imkânı oluşturulmuştur. 1964'lerde başlayan Kütüphane Otomasyonu, 20 yıl gibi kısa bir sürede inanılmaz boyutlara ulaşmıştır. Gelişmiş ülkelerde ve gelişmekte olan

¹²⁹ Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 8

¹³⁰ Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 9

ülkelerin pek çoğunda, artık Millî Bibliyografyalar bilgisayar yardımı ile üretilmektedir. Millî Bibliyografyalar için hazırlanan makine tarafından okunabilir katalog kayıtları diğer kütüphaneler tarafından kullanılmaktadır. Her büyüklükteki kütüphane, bir yandan kendisi için mümkün olan bilgisayar teknolojisini kullanırken, öte yandan daha büyük bilgisayarlara bağlanıp, diğer kütüphanelerin imkânlarından yararlanabilmektedir. Büyük bilgisayar donanımının maliyetini paylaşmak ve sağladığı hizmetten ortak yararlanmak üzere, yerel, kentsel, bölgesel, ulusal ve uluslararası Kütüphane Kooperatifleri ve Bilgi Ağları gelişmiştir. Ulusal ve uluslararası bilgi ve yayın alışverişi artık bilgisayar çıktısı olarak, bilgisayarca okunabilen ortamlarda, örneğin miknatıslı şerit veya bilgisayar-çıkışı-mikroform (Computer Output Microform-COM) biçiminde gerçekleştirilmekte; ya da iletişim teknolojisi kullanarak çevrim-içi (on-line) şeklinde doğrudan bilgisayar aracılığıyla yürütülmektedir. Kütüphane otomasyonu, teknolojinin tabiatı gereği, yeni örgütsel düzenlemeler getirmiştir ve gelişmiş ülkelerin hemen hepsi, 1970'li yıllar içinde, Ulusal Kütüphane ve Bilgi Akım Hizmetleri Örgütleri'ni yeniden biçimlendirme gereğini duymuşlar ve bunu gerçekleştirmişlerdir. Başta Birleşmiş Milletler ve bağlı ihtisas kuruluşları olmak üzere bütün uluslararası kuruluşlar, kütüphane ve bilgi akımında bilgisayar ve iletişim teknolojisi kullanmanın önemini vurgulamakta ve bu yönde faaliyet göstermektedirler. Özellikle UNESCO, üye ülkelere bu doğrultuda yardımcı olmaktadır. Dünyada kütüphane otomasyonu, kütüphanecilik işlemleri ve hizmetlerinde standardlaşmayı sağlamış ve bilgi akımını hızlandırmıştır.

2.7. Türkiye'de Kütüphane Otomasyonu Uygulamaları ¹³¹

Uygulama düzeyinde ilk kez 1971'de Hacettepe Üniversitesi Kütüphanesi süreli yayınlar katalogu bilgisayar çıktısı olarak basılmıştır. Bu çalışmada Dünya Sağlık Teşkilâtı Kütüphanesi süreli yayınlar katalogu örnek alınmıştır¹³². 1977 yılında Türkiye Bilimsel ve Teknik Araştırma Kurumu Dökümantasyon Merkezince (TÜRDÖK) yürütülen Ankara Süreli Yayınlar Toplu Katalogu'nun bilgisayar çıktısı olarak basılması gerçekleşmiştir. 1971'den günümüze kadar geçen bu zaman kesiti içinde gerek kütüphaneciler gerekse bilgisayar uzmanları konunun önemi, gerekliliği, yararlılığı ve yapılabilişliği konusunda olumlu bir fikir birliğine varmışlardır. Her iki meslek grubu konuyla yakından ilgilenmektedir. Bu sayede bilgisayar uzmanları ve kütüphaneciler birbirlerinin dillerinden anlar duruma gelmişlerdir. Türkiye'de kütüphanecilik hizmetlerinde¹³³ bilgisayar uygulamalarının gelişmesi bu iki grubun işbirliği ve ortak çalışmasına bağlıdır. Bu yönde gözlenen gelişmelerden birisi de eğitim alanındadır. Üniversitelerin kütüphanecilik bölümlerinde bilgisayar eğitime ilişkin dersler 1970'li yılların sonlarında başlamıştır. Bilgisayar eğitimi yapan bölümlerde tez çalışması ya da proje düzeyinde olmak üzere kütüphanecilik alanındaki çalışmalara ise yeni başladığı bu toplantıda dile getirilmiştir. Her iki durumda da uygulanan eğitim programlarının yeterli olduğu söylenemez. Ancak, olumlu bir gelişmenin ilk aşaması olarak kabul edilebilir.

¹³¹ Bkz. Leyla Tekin, Haziran 1996, Rektörlerin Gözüyle Üniversite Kütüphanelerinin Sorunları, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 10, Sayı 1, ISSN : 1300-0039, s. 166-169

¹³² Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 10-11

¹³³ Bkz. Doğan Atılgan, Mart 1995, Türkiye'de Kütüphaneciliğin Gelişimi, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 9, Sayı 1, ISSN : 1300-0039, s. 10-20

2.8. Türkiye'de Kütüphane Otomasyonunu Değerlendirme ¹³⁴

- 1- Türkiye'de Kütüphane Otomasyonu uygulaması, yetersiz düzeyde de olsa başlamış bulunmaktadır.
- 2- Ancak, bu uygulamalar birbirinden bağımsız, bireysel ve dağınıktır.
- 3- Kütüphane Otomasyonu uygulamaları çok yönlü, kapsamlı ve bütünleşik sistemler değildir. Genellikle sınırlı bir amaca yöneliktir.
- 4- Makina-İnsan ilişkisi ve iletişimine dayalı "interactive" bilgi erişim uygulaması yok denecek kadar azdır.
- 5- Uygulamayı yapan kuruluşlar arasında hiçbir ilişki, işbirliği ve ortak çalışma yoktur.
- 6- Bu durum, ileride bilgi ağları kurulma aşamasına gelindiğinde, sistemlerin birbiriyle uyumsuzluğu sonucunu doğuracak ve pek çok çalışmanın yenilenmesi gibi bir savurganlığa sebep olacaktır.
- 7- Uygulamalarda, gerek kütüphaneciler açısından (Örneğin; sınıflandırma, kataloglama, indeksleme v.b.) gerekse bilgisayar işlemleri açısından (format ve program çalışmaları) hiçbir standartlaşma yoktur. Her kuruluş kendi formatını kendisi geliştirmiş, kendi programını kendisi yazmıştır. Kütüphanecilik kuralları ise yine kuruluştan kuruluşa değişiklik arz etmektedir.
- 8- Standartlaşmanın olmayışı kütüphaneler arası bilgi değişimini engellemekte ve zorlaştırmaktadır. Oysa, Kütüphane Otomasyonunun başlıca amaçlarından birisi ulusal ve uluslararası bilgi değişimini ve akımını hızlandırmak ve bilgidan herkesin, dilediği anda ve hızda yararlanmasını sağlamaktır.
- 9- Kütüphane Otomasyonu alanında bir devlet politikası bulunmamaktadır. Hedefleri ve öncelikleri belirleyen; sorunları tesbit eden planlama, örgütlenme ve programlamayı yürüten; finansman kaynakları oluşturup işbirliği sağlayan bir devlet ya da hükümet programı bulunmadığı gibi, bu çalışmaları yürütecek ve denetleyecek bir ulusal kuruluş da mevcut değildir.
- 10- Ülkemizdeki kütüphane otomasyonu alanında her kütüphane kendi sorununu kendisi çözmeye uğraşmaktadır. Emek, donanım, para ve zaman açısından büyük savurganlık doğuran bu uygulama ülkemiz için son derece sakıncalıdır.
- 11- Diğer ülkelerde ve uluslararası düzeyde, Kütüphane Otomasyonu alanında ilk yatırım maliyetini en aza indirmek üzere ulusal ve uluslararası işbirliği ve ortak çalışma ve paylaşma ilkesine uyulmaktadır. Ülkemizde de savurganlığın önlenmesi, maliyetin ve imkânların paylaşılması ve etkin bir iletişim ağının kurulabilmesi için bu alanda işbirliği, ortak çalışma ve işbirliği zorunludur ve bunu gerçekleştirecek bir kuruma ihtiyaç vardır.
- 12- Bir ülkenin en etkin bilgi erişim araçlarından birisi olan Millî Bibliyografyalar'ın hazırlanması, yayınlanması ve kullanıma sunulmasında bilgisayar kullanımı, ülkemizde çok gecikmiştir.
- 13- Bu alanda insan gücü açığı vardır. Kütüphane Otomasyonu konusu, kütüphanecilik ve bilgisayar gibi iki ayrı uzmanlık alanının çakıştığı bir uygulamadır ve her iki grubun ortak çalışmasını gerektirir. Eğitim ve uygulama programlarında bu hususun gözetilmesi de son derece önemlidir.
- 14- Sonuç olarak; bilginin çabuk, doğru ve etkin bir biçimde iletişimini gerçekleştiren; ne biçimde, ne düzeyde ve nerede üretilirse üretilsin sınırlar ötesine taşıyan; bilgiyi tüm üretim sürecinin hizmetine, ihtiyaç duyulduğu anda getiren; evrensel bir çabanın ürünü

¹³⁴ Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 16-18

olan evrensel bilgiyi, bütün ülkelerde bütün kullanıcıların hizmetine sunmayı amaçlayan Kütüphane Otomasyonu alanında ülkemizde bazı çalışmaların ve uygulamaların yapılmakta olduğu görülmektedir. Ancak değindiğimiz gibi bu uygulamalar bireysel, dağınık ve yetersizdir ve ülkenin gerçek ihtiyacını karşılamaktan uzaktır¹³⁵.

Kütüphane Otomasyonu konusu, sadece bazı kütüphanelerde kitapların temini, kataloglanması ve raflara dizilmesi olarak ele alınmaktadır. Kütüphane Otomasyonu, öncelikle Türkiye'de olmak üzere dünya üzerinde; hangi ülkede, hangi konuda, hangi dilde, hangi düzeyde, ne biçimde ve nerede ne gibi bilgilerin varolduğunu belirleme; bu bilgilerin içeriğini tanımlama; hangi bilginin nereden nasıl ve ne zaman temin edilebileceğini tesbit etme; istenen bilgiden, istenildiği anda yararlanma imkânı sağlayan; ulusal ve uluslararası bilgi iletişimini etkinleştiren bir uygulama/çalışma alanı olarak değerlendirilmelidir. Uygulamalarının gerçekleşmesini ve yaygınlaşmasını engelleyen siyasal, yasal, ekonomik, teknolojik, örgütsel ve insan gücü sorunları vardır. Bu sorunları hiçbir kuruluş tek başına çözemez. Bu konu bir devlet politikası olarak ele alınmalı, ulusal düzeyde çözümler aranmalı ve geliştirilmelidir. Devlet düzeyinde destek ve örgütsel düzenlemeyi beklemeden, mevcut gelişmeler ve olanaklar değerlendirilmeli; uzmanlar düzeyinde çalışmalar yapılarak ortak etkinlik ve işbirliği ortamı oluşturulmalıdır. Bu öneriye uygun olarak, T.B.M.M. Kütüphanesinde bulunan bilgisayar donanımından yararlanmak suretiyle, anılan kütüphane ile diğer bazı kuruluşlarda başlatılmış olan program geliştirme çalışmalarından elde edilen sonuçları değerlendirmek ve ulusal düzeyde uygulanabilir bir standard format hazırlama çalışmasını yürütmek üzere Standard Format Çalışma Komitesi kurulması uygun görülmüştür. Komite beş kişiden oluşmaktadır. Anılan çalışma komitesinin gönüllü katkı ile değil, profesyonel bir statü ile çalışmasının sağlanması, komiteye süreklilik kazandırılması, ulusal bir örgüt kurulana kadar kütüphane ve bilgisayar uygulamaları bulunan resmi bir kuruma bağlı olarak görev yapması ve devletçe desteklenmesi zorunludur. Millî Kütüphane tarafından yayınlanmakta olan Millî Bibliyografyaların güncel, doğru ve etkin bir biçimde hazırlanması; hangi kütüphanelerde hangi kitaplar ve diğer belgelerin bulunduğunu gösteren toplu katalogların oluşturulması; ülkedeki bütün kütüphaneler için katalog kartları üretecek bir merkezi kataloglama ve kart dağıtım sistemi geliştirilmesi için Millî Kütüphane'de bilgisayar uygulama çalışmaları desteklenmeli ve hızlandırılmalıdır¹³⁶.

Kütüphane Otomasyonu, bilgisayar uzmanları ile kütüphane uzmanlarının bir arada çalışmasını gerektiren bir konudur. Bu nedenle, üniversitelerin bilgisayar bölümlerinde kütüphanecilik bilgileri; kütüphanecilik bölümlerinde ise bilgisayar bilgileri daha yoğun ve yaygın hale getirilmelidir. Halen çeşitli kütüphanelerde görev yapmakta olan kütüphanecilerin kütüphane otomasyonu alanında bilgilendirilmesi ve yetiştirilmesi için kurslar düzenlenmelidir. Kütüphane Otomasyonunun yaygınlaştırılması amacına yönelik olarak kamuoyu oluşturmaya ve bilinçlendirmeye yönelik seminer ve benzeri toplantılar düzenlenmelidir. Bilgisayar aracılığı ile bilgi değişiminde zaman ve emek tasarrufu sağlayan; bilgi değişimini, tüm katalog bilgilerine gerek olmaksızın, bütün dünya

¹³⁵ Türkiye'deki Üniversite Kütüphaneleri ile ilgili ayrıntılı bilgi almak isteyenler İnternet üzerinden <http://www.ulakbim.gov.tr/hizmetler/> adresine erişebilirler.

¹³⁶ Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 18-19

kütüphanecilerinin anlayacağı biçimde bir tek numara ile gerçekleştirmeye yarayan Uluslararası Standard Kitap Numarası (International Standard Book Number-ISBN¹³⁷) ile Uluslararası Standard Süreli Yayın Numarası (International Standard Serial Number-ISSN) uygulaması gerçekleştirilmelidir. Kütüphane Otomasyonu alanında dünyadaki gelişmeler ve eğilimler yakından izlenmeli; özellikle uluslararası kuruluşlar tarafından geliştirilmekte olan uluslararası bilgi ve kütüphane sistemlerinden yararlanma yolları aranmalıdır. Ayrıca uluslararası düzeyde benimsenmiş olan standartlarla uyum sağlanmalıdır. Kamu kuruluşlarında mevcut büyük bilgisayar sistemlerinin boş kapasiteleri, eleman takviyesi ile Kütüphane Otomasyonu alanında değerlendirilmelidir. Kütüphane otomasyonunun ülke düzeyinde, belirli bir sistem yaklaşımı içinde geliştirilmesi ve yaygınlaştırılması doğrultusunda ulusal düzeyde çalışmalar yapılmalı; desteklenmelidir. Bu çalışmalar, Beş-Yıllık Kalkınma Plânları ile Yıllık Programlarda yer alacak tedbirlerle zorlanmalı ve teşvik edilmelidir. Kütüphane otomasyonu alanında, ulusal düzeyde, araştırma/planlama ve işbirliği çalışmalarını yürütecek; bu alandaki uygulamaları izleyecek ve denetleyecek ulusal bir örgüt kurulmalıdır. Önerilerin gerçekleştirilmesi ve kütüphane otomasyonu alanında gereken çalışmaların yapılabilmesi için böyle bir kuruluşa acilen ihtiyaç duyulmaktadır¹³⁸.

2.9. Uygulama Geliştirmede Kullanılan Araçlar

2.9.1. SQL/DS (Structured Query Language / Data System)

SQL¹³⁹, veritabanı uygulamalarında verilerin tanımlanması, veritabanının bütünlüğünün ve veritabanına erişimin kontrolü, çeşitli kriterlere göre bilgi/rapor elde edilmesi, veritabanlarının sorgulanması ve güncellenmesi amaçları için gerekli komutlara sahip olan bir alt dildir. Alt dil denmesinin nedeni bir bilgisayar dilinin sahip olması gereken tüm

¹³⁷ Yabancı kitaplarda 1967'den bu yana yer alan 10 basamaklı bir numaralama sistemidir. Bu standart ilk kez ISO (International Standard Organization) tarafından ISO/2100 sayı ve 1972 tarihle yayınlanmış, Türk Standartlar Enstitüsü Teknik Kurulu'na da onaylanarak TS 2143/Kasım 1975 sayı ve tarihle Türkçe olarak yayınlanmıştır. ISBN dört gruptan oluşmaktadır :

1-Grup tanıtıcı : Yayınevinin bulunduğu ülke, dil v.b. gösterir.

2-Yayınevi tanıtıcı : Grup içindeki yayınevi ya da yayıncıyı gösterir.

3-Başlık tanıtıcı : Bu grupta kitap işareti yer almaktadır ve ilk işarete göre uzunluğu tesbit edilmektedir.

4-Denetim sayısı : Verilen ISBN numarasının doğruluğunu sağlar ve tek rakamlıdır.

Türkiye'de ISBN numarası veren ilk yayın Sekine Karakaş ve Hasan Duman tarafından yayınlanan "Kitap numaralama sistemi el kitabı ve Türkiye'de uygulaması" isimli eserdir. 1987'den itibaren Türkiye'de ISBN numarası verme çalışmaları başlatılmıştır. (Hasan S. Keseroğlu, 1988, Kataloqlama Kuralları Örnekleri ile Anglo-Amerikan Kataloqlama Kuralları, Edebiyat Fakültesi Basımevi, İstanbul, s. 84-85)

¹³⁸ Anonymous, Türkiye'de Kütüphanecilik Alanında....., a.g.e., s. 20

¹³⁹ İlişkisel veritabanlarının prensipleri ilk olarak Edgar F. Codd'un (IBM) tarafından 1970'de yayınlanan "A Relational Model for Large Shared Data Banks-Geniş Paylaşımlı Veri Bankaları için İlişkisel Veri Modeli" adlı makalesine dayanmaktadır. Bkz. E.F.Codd, A Relational Model for Large Shared Data Banks, June 1970, Communications of the ACM, Volume 13, Number 6, p. 377-387. Ayrıca Bkz. E.F.Codd, 1972, Further Normalization of the Data Base Relational Model, in Rustin, p. 33-64.

komutlara sahip olamayışı sebebi iledir¹⁴⁰. İlk SQL standardı 1987’de ISO 9075 kodu ile yayınlanmıştır. SQL¹⁴¹ ilişkisel veritabanının özellikleri ise şunlardır:

1-Bilginin tablolar halinde tutulması ve görüntülenmesi : SQL ortamında bilgiler tablolar halinde tutulmakta ve görüntülenmektedir. Tablolar sıra ve kolonlardan meydana gelmekte, bilgiye ulaşmak istenildiği zaman kolon isimlerinin belirtilmesi yeterli olmaktadır.

2-Sonuca yönelik ulaşım : SQL ortamındaki bilgilere çok basit komutlar aracılığı ile erişilmektedir. Bu işlemi yaparken “bilgiye nasıl ulaşacağım ?” diye bir problem yoktur. Kullanıcı istediği bilgiyi SQL komutları aracılığı ile belirtmekte bilgiye erişimi SQL sağlamaktadır.

3-Dinamik erişim : Tablolar arasındaki dinamik ilişki, iki tablonun en az bir kolonunun içeriğinin aynı olması sayesinde sağlanmaktadır. Bu durumda iki tablo birleşmektedir. Fakat bu birleşme tamamen mantıksal olarak yapılmakta, fiziki bir birleşim söz konusu olmamaktadır.

4-Otomatik optimizasyon : SQL bilgilere ulaşırken, ulaşacağı yolu çeşitli optimizasyon teknikleri kullanarak kendisi seçmektedir.

SQL komutları kullanıcıların kolay kullanabilmeleri için konuşma diline yakın tasarlanmıştır. SQL etkileşimli olarak da yani kullanıcı terminal başına oturup program yazmadan, sadece SQL sorgularını girerek veritabanına ulaşabilmekte, SQL’in bu tür kullanımına ISQL (Interactive Structured Query Language) denmektedir. SQL komutlarının hemen hemen hepsi ISQL ortamında da kullanılabilir. ISQL özellikle plânlanmış sorgulamaların kullanımında çok etkilidir ve birtakım özel işlemler de yapabilmektedir. ISQL ortamında çok sık kullanılan sorgular istenirse bir isim verilerek saklanabilmektedir. Bunun iki amacı vardır. Birincisi, yazılması uzun süren sorgularda her defasında uzun uzun yazmak yerine bir kez yazıp saklamak ve gerektiğinde kullanmak, ikincisi sorgu hazırlanırken bir takım parametreleri dışarıdan verebilmek için. Böylece aynı sorgu değişik bilgiler elde etmede kullanılabilir. ISQL’in saklanmış sorgular ve format komutlarının birleşiminden oluşan “routine” adlı bir tablosu vardır. Bu tabloya, “select” komutu ve peşinden de “format” komutları girilerek tabloda belirli bir isim altında, bir grup komut oluşturulmakta ve bu komutlar topluluğu “run” komutuyla çalıştırılabilmektedir. Routine de bir tablo olduğu için komutlar tabloya “insert” veya “input” komutlarıyla ilave edilebilmektedir¹⁴².

¹⁴⁰ Prof. Dr. Mithat Uysal, Aralık 1994, SQL Veritabanı Sorgulama Dili, Beta Basım Yayın Dağıtım A.Ş., İstanbul, ISBN : 975-486-389-9, s. IX

¹⁴¹ Bkz. IBM, December 1988, Managing SQL/Data System, Version 2 Release 2, SH09-8069-01, U.S.A., p. 1-18

¹⁴² Anonymous, IBM Türk Ltd. Şti. Eğ. Müd., Mayıs 1988, SQL/DS Tanıtma ve Uygulama Geliştirme Kurs Notları, s. 3-4

ISQL ortamında F1 tuşuna basıldığında bir yardım menüsü ekrana gelmekte, burada ya bir seçim yapılarak ya da “help komut adı” girilerek belirli bir komutun yazılım kuralları hakkındaki bilgilere erişilebilmektedir. SQL komutları, bağımsız kullanılabildikleri gibi bir programla dili ile beraber de kullanılabilmektedir. Destekleyici programlardan bazıları Cobol, Assembler, Fortran ve PL/1’dir. Komut sonlarında PL/1’da “;”, Cobol’da “END-EXEC” gelmeli, SQL komutlarının başına ise EXEC.SQL konulmalıdır. Program, ön derleyiciden geçerken EXEC.SQL deyimleri, kullanılan programlama dilinin anlayabileceği CALL deyimine dönüşmektedir.

SQL’nin yaptığı belli başlı işlerden biri de, bir tabloda belirli sıraları işlem yapmak üzere seçmektir. Seçilen sıralar belirli bir arama şartına (search condition) bağlı olarak seçilmekte, bir şart ise bir veya daha fazla ifadeden (predicate) oluşmaktadır. Her bir ifade SQL için bir test işlemi demektir. İfadeler, mantıksal operatörler olan AND-OR-NOT ile birbirlerine bağlanır. Parantezler kullanılarak işlem önceliği (işlem önceliği Not-And-Or’dur) değiştirilebilmektedir. Veritabanından birden fazla satır çekilmek istendiğinde “cursor” adı verilen komut kullanılmakta, dönen bilgiye “aktif set” denilmektedir.

Görüntüler (views) ise çeşitli kullanıcıların aynı veriyi değişik şekillerde görmelerine yarar SQL’in diğer bir imkanındır. Bir veya birden fazla tablodan her kullanıcıyı ilgilendiren bilgiler ayrılarak bir görüntü oluşturulmakta ve kullanıcı bu görüntüye ulaşmaktadır. Her görüntü aynen bir tablo gibi olup bir adı vardır. Daha çok güvenlik amacıyla kullanılmaktadır. Görüntüler tablolar üzerinde dinamik bir pencere gibidir. Bu şu anlama gelmektedir. Tablo üzerinde bir “update” işlemi yapıldığında görüntü de aynen bu işleminden etkilenmektedir. Bunun tam tersi de geçerlidir. Yani bir görüntü “update” işlemine maruz kalırsa görüntü arkasındaki tablo da güncellenmektedir. Bu konuda bazı sınırlamalar vardır. Bunun sebebi SQL’in görüntüleri fiziksel olarak saklamayıp, görüntüler üzerine indeksler oluşturmadığından dolayıdır. Fakat tablo üzerinde bir indeks oluşturulduğunda görüntüdeki performans da artmaktadır. Her uygulamanın başına bir takım komutlar (application prologue) konarak SQL iletişim alanını (SQLCA) tanımlamak, hata durumlarını test etmek, host değişkenleri tanımlamak ayrıca SQL ile program arasında iletişim kurmak için kullanılabilmektedir. SQLCA iletişim alanını tanımlamak için program içerisine “Exec Sql Include Sqlca” komutu konulmalıdır. Program ön derlemeyden geçerken SQL host değişkenlerini bu alana eklemektedir. Bu değişken grubu hataların belirlenmesi ve uyarıların yapılmasını sağlamaktadır. Her SQL komutu yürütüldükten sonra bir SQL dönüş kodu (SQLCODE) üretmektedir. Komut normal olarak yürütülmüşse bu kod 0’dır. Hata durumunda SQL’in döndürdüğü kod negatiftir. Beklenmeyen durumlarda hata dönüş kodu pozitifdir. Negatif hata dönüş kodlarında ne yapılması gerektiği “Whenever” komutu ile belirtilebilmektedir¹⁴³.

SQL, her bir kullanıcının hangi yetkilere sahip olduğunu kaydını da tutmaktadır. SQL kurulduğunda en az bir kişiye DBA (Database Administor) yetkisi verilmelidir. Bu kişi bütün SQL kaynaklarının kullanımı kontrol etmekte, kullanıcılara gerekli yetkileri vermektedir. Veritabanı ile ilgili bir işlem yapmadan önce SQL’i kullanma yetkisine sahip

¹⁴³ Anonymous, IBM Türk Ltd., a.g.e., s. 6-21

olmak gereklidir. Bu özel yetki Connect yetkisidir ve DBA tarafından verilmektedir. DBA bu yetkiyi tüm kullanıcılara verebilir. Yetkiler üç grup altında incelenmektedir.

1-Tablo ve görüntülerdeki ayrıcalıklar : Kullanıcı kendi alanında bir tablo oluşturduğu zaman otomatik olarak bütün yetkilere sahip olmaktadır. SQL her yetki için "Grant" seçeneğini bu kullanıcıya verir. Kullanıcı da bu yetkileri yine "Grant" komutu ile başkalarına verebilmektedir. Bir kullanıcıya verilen yetki geri alındığında onun yetki verdiği tüm kullanıcılardan da alınmış olur. "Alter" ve "Index" dışındaki tüm yetkiler görüntüler için de geçerlidir. Tablo ve görüntüler için verilen yetkiler SQL kataloğunda olan "Systabauth" ve "Syscolauth" tablolarında tutulmaktadır.

2-Program ayrıcalıkları : Bütün SQL uygulama programları bir ön derlemeden geçmek zorundadır. Ön derleyici bir erişim modülü oluşturup bunu veritabanında saklamaktadır. Bir program, ön derlemeden hatasız olarak geçtiğinde "Run" ayrıcalığına sahip olmaktadır. Bu da programı istediğiniz zaman çalıştırabileceğiniz anlamına gelmektedir. SQL kataloglarından "Sysprogauth" tablosunda "Run" yetkisine sahip kullanıcılar ve programların listesi tutulmaktadır.

3-Özel ayrıcalıklar : Tablo ve programlar için yetkilerden başka "connect, resource ve dba" yetkileri de bulunmaktadır. Connect yetkisi, SQL kataloglarında kullanıcı için bir kaydın var olduğunu göstermektedir. Resource yetkisi kullanıcılara public dbspace'ler içinde tablo oluşturma ve private dbspace alma (acquire) yetkisi verir. Private dbspace'i olan bir kişinin tablo oluşturma için bu yetkiye ihtiyacı yoktur. DbA yetkisine sahip bir kişi otomatik olarak "resource" ve "connect" yetkilerine sahip olmaktadır. Sistemin en yüksek yetkisi DbA yetkisidir.

Özel ayrıcalıklar "Sysuserauth" katalogunda tutulmakta ve bu kataloga sadece "DbA" yetkisi bulunan kişiler ulaşabilmektedir. Diğer kişiler bu tablodan elde edilen görüntü olan "Sysuserlist" tablosunda sorgu yapabilmektedirler. Görüntüde şifreler görülmemektedir¹⁴⁴.

2.9.1.1. SQL Veritabanının Yapısı

SQL veritabanı, bir VM makinasıdır. Bir veya daha fazla veritabanından oluşabilmektedir. Veritabanı makinası belirli bir zaman için yalnızca bir veritabanı ile aktif olur. SQL makinası "sqlstart exec" ile başlatılır ve "sqlend" komutu ile kapatılır. Başlarken verilen "dbname" parametresi hangi veritabanını harekete geçireceğini belirtir. VM işletim sisteminde açıkça belirtilen "connect" komutu desteklenir. Fakat bu komutu kullanmak mecburi değildir. Komut verilmez ise SQL, VM kullanıcı tanımını (userid) SQL kullanıcı tanımı olarak kabul eder. Bu da SQL'in üstü kapalı "connect" komutunu desteklediği anlamına gelmektedir ve ya özel bir kullanıcı adı olan Allusers'a "connect" yetkisi verilmeli ya da kullanıcıya "connect" yetkisi verilmelidir.

SQL'in belirtilen veritabanı ile kullanıcı makinası arasında ilişki kurması için bir defa çalıştırılması gereken bir program vardır. "Sqlinit exec." Çok kullanıcıli durumlarda bu iş

¹⁴⁴ Anonymous, IBM Türk Ltd. ..., a.g.e., s. 30-31

için SQL makinasının belirtilen veritabanı ile başlatılmış olması gerekir. Tek kullanıcı durumunda SQL'i başlatma görevi de "sqlerp" programına aittir. Uygulamalar hem tek hem de çok kullanıcı durumunda çalıştırabilmektedir. Her iki durumda da SQL'in üretim minidiskine (Q) erişmek gerekmektedir.

Tabloların ve bunlarla ilgili izinlerin veritabanı içerisinde durduğu alanlara "dbspace" adı verilmektedir ve tabloyu oluşturan tarafından belli bir dbspace'e konması gerekmektedir. Dbspace'ler veritabanı oluştururken tanımlanabilmekte veya daha sonra "add dbspace" komutu ile de eklenebilmektedirler. Her dbspace, "acquire dbspace" komutu yürütülene kadar isimsiz ve boştur. Bu iş için -genellikle DBA- dbspace'i hangi bellek havuzundan (storage pool) alacağını belirtir. Belirtmezse SQL tarafından varsayılan yer (default) geçerlidir. Bellek havuzu dbextent'lerden oluşan bilgi takımları olup 1-999 arasında bir değer alabilmektedir. Görevleri diskler üzerindeki bilginin dağılımını denetlemektir. Bellek havuzları kurtarılabilen (recoverable) veya kurtarılamayan (nonrecoverable) olmak üzere iki türde olmaktadır. Kurtarılabilen türdeki bellek havuzları, üzerlerindeki bilgiyi herhangi bir hataya karşı koruyabilmektedirler. Sahip olunan dbspace'in tipi "private" ise sadece tek bir kişi tarafından erişilmekte, "public" ise herkes tarafından erişilebilmektedir. Bir dbspace, "Drop Dbspace" komutu ile silinebilmektedir. "Resource" yetkisine sahip biri public dbspace'lerde veya kendine ait private dbspace'de tablolar oluşturabilmektedir. "Dba" yetkisine sahip kişiler başkasına ait private dbspace'lerde de tablo oluşturabilmekte, başka birinin dbspace'ine erişilebilmektedirler. Fakat bu dbspace kullanılıyorsa buna izin verilmemektedir.

2.9.1.2. Mantıksal İş Birimi (Logical Unit of Work-LUW)

SQL tarafından bir adım olarak değerlendirilen SQL komutlarına mantıksal iş birimi denmektedir. Sql bilgi bütünlüğünü bir mantıksal iş birimi içinde ele almaktadır ki bu da ya bütün işlemlerin tümünü doğru bir biçimde tamamlanır veya hiçbiri yapılmaz demektir. Örneğin bir hesaptan bir miktar para alınıp başka hesaba yatırılacaksa ve de bu iki işlem aynı mantıksal birim içindeyse bilgide herhangi bir tutarsızlık olmayacaktır. Sistemde bir arıza olması halinde hiç bir iş yapılmamış gibi değerlendirilecektir. Bir mantıksal iş birimi -connect haricinde- herhangi bir yürütülebilir SQL deyimi görüldüğünde başlamakta, "commit work" veya "rollback work" görüldüğünde sona ermektedir. Eğer bu komutlara rastlanmazsa program bittiğinde mantıksal birim de biter. SQL tanımlama komutları (begin dcl, whenver v.b.) bir luw başlatmaz. "Commit work" deyimi luw birimini sona erdirmektedir.

"Rollback work", luw birimini başlangıç durumuna getirir ve mantıksal birimi bitirmektedir. Eğer bir hata sebebiyle girilen bir rutinde rollback work kullanılacaksa daha önce "whenewer sqlerror continue" veya "whenewer sqlwarning continue" komutları verilerek rollback work komutunda problem olursa programın döngüye girmesi engellenmelidir¹⁴⁵.

Tablo oluşturmak için resource yetkisine sahip olunması gerekmektedir. Bir kolona "not null" seçeneği konulursa SQL bu kolonda boş bir değer bulunmasına izin vermez. Her

¹⁴⁵ Anonymous, IBM Türk Ltd. ..., a.g.e., s. 33-48

tablonun bir sahibi vardır ve bu bilgi SQL tablolarında tutulmaktadır. Yeni oluşturulan bir tablo aşağıdaki kurallar çerçevesinde varolan dbspace'lerden birine yerleşmektedir.

1- "Create table" komutunda bir dbspace ismi verilmişse tablo o dbspace'e yerleşir. Dbspace'in sahibi ya programı ön derleyiciden geçiren veya public olmalıdır. DBA yetkisi varsa herhangi bir kullanıcının dbspace'in üzerinde onun adı belirtilerek tablo oluşturulabilir.

Örnek : Create table X in Ahmet DBB

2- "Create table" komutunda dbspace adı belirtilmemişse tablo programı ön derlemeden geçer ve kullanıcının private dbspace'ine yerleşir. Böyle bir dbspace yoksa hata oluşur.

3- Eğer tablo "non-recoverable" dbspace'de oluşturmak isteniyorsa mutlaka dbspace adı verilmelidir.

2.9.1.3. SQL'nin Otomatik Kilitleme Mekanizması

SQL'iki ayrı durumda çalışabilmektedir. Bunlar tek (single) ve çok kullanıcı (multiple) durumlarıdır. Tek kullanıcı durumunda çalışırken başkaları sizin kullandığınız bilgiyi kullanamamakta, çok kullanıcı durumunda ise aynı bilgi değişik kullanıcılar arasında paylaşılmaktadır. SQL'e aynı anda erişildiğinde çıkacak problemler için otomatik kilitleme mekanizması vardır. Her bir mantıksal iş biriminde erişilen bilgi SQL tarafından kilitlenir. Kilitleme paylaşımlı (share) ve paylaşımsız (exclusive) olmak üzere iki çeşittir. Paylaşımsız kilitleme diğer kullanıcıların görmesini ve değiştirmesini engellerken, paylaşımlı kilitleme diğer kullanıcıların görmesini serbest bırakıp, değiştirilmesini engellemektedir. Kilitler her "luw" sonucunda serbest bırakılırlar. Bu sebeple programlarda her SQL komutunun önüne ve arkasına kitlemeleri önlemek için "commit work" koymak gereklidir. SQL, karşılıklı kilitleme (deadlock) durumlarını da denetlemekte ve kendi kendine çözmektedir. Karşılıklı kilitleme, iki mantıksal iş biriminin diğeri tarafından kilitlenmiş kaynağı kullanmak istemesidir. Bu durumda SQL en son gelen işi geriye çekerek (back-out) yaptığı bütün değişiklikleri geri almakta ve kilitleme mekanizmasını serbest bırakmaktadır.

Bir dbspace'i kitlemek için ya dbspace'in sahibi veya dba yetkisine sahip olmak gerekir. Bir tablonun sahibi veya o tablo üzerinde de "select" veya "dba" yetkisi olan kişiler o tabloyu kitleyebilmektedir. Ancak ne tür yetkiye sahip olursa olsun hiç bir kullanıcı SQL/DS katalogunu veya katalogların bulunduğu dbspace'i kitleyemez. "Lock" komutu ise istenen tablo veya dbspace üzerinde tümüyle bir kitleme sağlar. O anki "luw" bitene kadar bu kitleme tutulur. Çok kullanıcı modda çok yararlı bir komuttur¹⁴⁶.

2.9.1.4. İndeksler

İndeks oluşturulacak kolon "long varchar" veya "long vargraphic" olmamalıdır. İndeks artan (ascending) veya azalan (descending) bir sırada oluşturabilmektedir. Varsayılan değer artan sıradadır. "Drop index" komutu veya bağlı olduğu dbspace silinene kadar indeksin bakımı SQL tarafından yapılmaktadır. İndeksler uygulama programlarını

¹⁴⁶ Anonymous, IBM Türk Ltd. ..., a.g.e., s. 37-38

etkilememektedir. SQL'in ön derleyicisi sorgu sırasında hangi indeksi kullanacağına kendisi karar vermektedir. İndeks kullanırken bazı noktalara dikkat edilmelidir:

- 1- Bir kolon üzerine oluşturulan indeks bu tabloya o kolondan erişimde kolaylık sağlamaktadır, fakat bu tablonun o kolonu güncelleştirilirken cevap süresi (response time) uzamaktadır. Sebebi o kolonun indeks alanlarının da düzenlenmesidir.
- 2- Bir kolon üzerinde tek (unique) bir indeks oluşturulursa o kolondaki iki sıranın aynı bilgiye sahip olmaması gerekmektedir. Çift kayıt (duplicate entry) bulunan kolonda unique indeks oluşturulmaya kalkışılırsa SQL bir hata kodu üretecektir. "Unique" durumunu bozacak herhangi bir "insert" veya "update" işlemi hata ile sonuçlanacaktır. "Create index" komutundaki "pctfree" parametresi dizin için ayrılan alanların % olarak ne kadarının sonraki insert'ler için boş kalacağını belirtmektedir. % 50'yi geçmemesi tavsiye edilmektedir. Bir indeks en fazla en fazla 16 kolon üzerine yapılabilir. İndeks, tablo bilgileri yüklendikten sonra oluşturuluyorsa SQL sıralama rutini "sort" harekete geçmektedir. Eğer çift kayıt yapılmasına izin varsa sıralama rutini 4 byte fazladan yer istemektedir.

Yeni bir indeks oluşturulurken dikkat edilecek hususlar ise şunlardır.

- 1- Program ön derleyiciden geçtiğinde en iyi erişim yolunu bulan bir erişim modülü oluşturduğu için pratikte, indeksleri oluşturduktan sonra derlemek daha yararlı olmaktadır.
- 2- Erişim modülleri oluşmuş programlarda, yeni bir indeks tanımlandığı zaman, erişim modülleri geçersiz (invalid) olmamakta, onlar hâla kendi erişim yollarını (path) kullanmaktadırlar. Bu sebeple indeks oluşturduktan sonra performansı artırmak için programı yeniden derlemek daha faydalı olmaktadır. Bu durumda yeni indeksi de göz önüne alan yeni bir erişim modülü oluşturulacaktır.

2.9.1.5. SQL Katalogları

SQL, veritabanı ile ilgili bilgilerin bakımını katalog adı verilen tablolarda yapmaktadır. Veritabanı oluşturulurken bu tablolar otomatik olarak oluşturulmaktadır. Bu tablolarda oluşturulan tablolar, kolonlar, bu tablolara ait indeksler, programlar, yetkileri ve diğer bilgiler tutulmaktadır. Bu tablolar herkese açık olup (public), herkese bunları okuma yetkisine sahiptir. Bütün katalog tablolarının sahibi sistemdir. Bu sebeple "system" ön eki katalog tablolarına erişirken belirtilmelidir. Parolaları içeren tabloya ulaşmak için "dba" yetkisine sahip olmak gerekir¹⁴⁷.

2.9.1.6. SQL/DBSU (Database System Utility - Veritabanı Hizmet Programı)

SQL tablolarına toplu olarak bilgi yüklemek amacıyla "datalog" komutu kullanılmaktadır. Ayrıca "dataunload" komutuyla yedeklenmiş bilgiler de "datalog" komutuyla yüklenebilmektedir. Bu komuttaki her giriş kaydı, tabloda bir sıra oluşturmaktadır. Bu giriş kayıtları sıralı bir kütük olabilir veya kontrol komutlarının arkasına yerleştirilir. Bilgi yüklenen tablolar daha önceden bilgi içeriyor olabilir. Bu

¹⁴⁷ Anonymous, IBM Türk Ltd., a.g.e., s. 41-46

durumda eski bilgiler yenilerden etkilenmez. Ayrıca tablo üzerinde mevcut bir indeks varsa bu indeks gelen bilgilere göre yeniden düzenlenmektedir¹⁴⁸.

2.9.2. CSP (Cross System Product)

CSP¹⁴⁹ son kullanıcılara hitap eden uygulamalarda mümkün olduğunca çabuk ve yüksek kalitede hizmet vermeyi sağlayan etkileşimli (interactive) bir program setidir. CSP/Application Development (CSP/AD) ve CSP/Application Execution (CSP/AE) CSP'nin iki önemli bileşenidir.

CSP program setinde tanımlama, test, derleme ve uygulama programlarını çalıştırma işlemleri tamamen ilgili elemanlarca yapılabilmektedir. Program setinin elemanları kullanıcılara zamandan, organizasyon için de maliyetlerden tasarruf sağlayarak çoklu gelişim ve çoklu ürün çevrelerini desteklemektedir. Uygulama geliştirme devresinde ise CSP program seti aşağıdaki potansiyel faydaları temin etmektedir.

1-Verimliliğin artırılması : Diğer geleneksel uygulama geliştirme metodları ile karşılaştırıldığında CSP'nin, verimliliği anlamlı bir şekilde artırdığı görülür. Cobol ve CSP/AD kullanan iki gruptan, CSP kullanan grubun CSP/AD'nin daha az kodlama gerektirmesi sebebi ile daha etkin bir uygulama geliştirme programı kullandığı söylenebilir.

2-Bütünlüğü sağlamak : Uygulamaların yukarıdan-aşağıya (top-down¹⁵⁰) doğru dizayn edilmesi, standart giriş-çıkış (input-output) işlemleri ve ifadelerin kontrolü bütünlük sağlayan unsurlardır. Ekran dizaynları veya ekranlar (maps), uygulama listeleri daima aynı mantıkla tanımlanmaktadır. CSP'nin sağladığı bu bütünlük sayesinde uygulamaların bakımı kolaylaşmakta ve maliyetleri azaltıcı bir etki yapmaktadır¹⁵¹.

3-Taşınabilirlik : CSP/AD tanımları, desteklediği değişik ortamlara taşınabilmektedir. Veriler, paneller ve uygulamaya ait tanımlar MSL'den bir dosyaya alınarak bir sistemden diğerine taşınabilmektedir. Uygulamalar mutlaka CSP/AD tanımları ve işletim ifadeleri ile tek başına tanımlanmalı ve fonksiyon bağımsız herhangi bir sistem kullanılmamalıdır. Derlenen uygulamalar desteklenen -aynı veya daha yüksek seviyedeki- herhangi bir ortamda, CSP/AE ise aynı veya daha düşük bir CSP/AD seviyesinde derlenerek çalıştırılabilmektedir¹⁵². CSP/AD'nin çapraz sistem desteği herhangi bir sistem için yazılan ve geliştirilen bir uygulamanın diğer sistemlerde de çalıştırılmasına imkân tanımaktadır. Örneğin, bir uygulama TSO sistemi altında geliştirilip CICS/MVS sistemine taşınarak çalıştırılabilir. CSP ürün seti, üzerinde çalışılan işletim sistemi ile altındaki donanım arasındaki farklılığı gizleyerek, CSP/AD uygulamalarının bir işletim sisteminden diğerine

¹⁴⁸ Anonymous, IBM Türk Ltd. ..., a.g.e., s. 50

¹⁴⁹ Ayr. Bkz. Meltem Uğur, April 1991, Use of CSP/AD and CSP/AE on IBM 3090/180S, METU Computer Center Bulletin (CCNEWS), Document No : CC-91.01, Ankara, Turkey, p. 1-3 ve W. K. Haynes, M. E. Dwell, P.J. Herman, 1988, The Cross system Product Application Generator : An Evolution, IBM System Journals, Vol. 27, No 3, U.S.A., ISSN : 18-8670, p. 384-390

¹⁵⁰ C. Gane and T. Sarson, a.g.e., p. 216

¹⁵¹ Anonymous, 1990, Cross System Product General Information, U.S.A., GH23-0500-4, s. 1

¹⁵² Anonymous, Cross System..., a.g.e., s. 44

üzerinde pek az veya hiç bir değişiklik yapılmaksızın taşınmasını sağlamaktadır. Çok görevliliğin bulunduğu çevreler, CSP ürün setini, çok yönlü bir uygulama derleyicisi meydana getirerek, CSP ürün seti ise, çok görevliliğe sahip ortamları, çok yönlü bir uygulama derleyicisi oluşturarak desteklemektedir. CSP/AD program geliştirme ve test çevresi herhangi bir IBM System/370 işlemci kullanan ve IBM tarafından geliştirilen herhangi bir ana işletim çevresi de olabilir. CSP/AE ve CSP/370 Runtime Services ise desteklenen ortamlarda çalışan uygulamaların taşınabilirliğini temin etmektedir. CSP/AE kullanılarak bir uygulama desteklenen sistemlerde geliştirilip çalıştırılabilir ve CSP/AD'den bağımsız çalışır. Örneğin, MVS/TSO sisteminde geliştirilip test edilen uygulamalar bir CICS makinası veya DPPX sistemde çalıştırılabilir. CSP/AD Programmable Workstation Feature (PWF), OS/2 Presentation Manager kullanan bir IBM PS/2 ortamında da uygulama geliştirmeye elverişli şekilde tasarlanmıştır. IBM System/370 üzerinde CSP/AD, IBM PS/2'de ise CSP/AD PWF kullanılarak uygulamalar tanımlanabilmekte ve yönetilebilmektedir. CSP/Application Generation EZ-PREP ve CSP/Application Execution-EZ-RUN ile CSP uygulamaları IBM DOS veya bir PS/2'de çalıştırılabilmektedir. Uygulamalar Application System/400 kullanılarak da derlenebilmektedir. CSP'nin çapraz sistem niteliği uygulama geliştirenlerin ayrı ayrı işletim ortamlarında önemli birikimler meydana getirmelerini sağlar¹⁵³.

4-Kullanım kolaylığı : Profesyonel veri işlemlerde, programlama görevleri uygulama araçlarının ve mevcut yüksek-seviyeli programlama dillerinin geniş bir aralığı kullanılarak yapılmaktadır. Tek bir ürün seti olan CSP programı ise bu araçların bir benzerini kullanarak tanımlama, test ve derleme işlemlerinin yapılmasına imkân vermektedir. Kodlama işlemleri azaltılarak uygulama geliştirme işlemini basitleştirilmekte, uygulamaya ait tanımların yapısı ve CSP/AD'nin uzmanlaşmış dili sayesinde elverişli uygulama programlarının geliştirilmesini sağlanmaktadır. Uygulama geliştirme programlarının çok azı, zamandan ve geliştirme maliyetlerinden tasarruf sağlamaktadır. CSP/AD PWF, bunlardan da öte uygulama geliştirmeyi, terminal veya iş istasyonlarının gücünü ve özelliklerini kullanarak daha da basitleştirmektedir. Geliştirmenin yapıldığı çevrede etkileşimli ve uygun bir arabirimle, kullanım kolaylığı ve tek bir biçim oluşturulmakta, grafik imkânları tanımlanan uygulamanın yapısını görülebilecek şekilde sunmakta, bu da yazılım geliştirenlerin uygulama yapısını anlamalarına yardımcı olmaktadır. Geleneksel metotlarla meydana getirilen uygulama programlarının sık sık ihtiyaç duyduğu çoğu adımları CSP/AD en aza indirmektedir. CSP seti ile projenin her bir bölümü; etkileşimli ekranların tanımlanması ve doğrulanması, mantık testleri ve hata ayıklama programları, uygulamaların deneme-yanılma çalışmalarının başlatılması ve programın uygulama içine konması adımlarından meydana gelmektedir. Ayrıca;

- 1- Dialog yönetimi ile, uygulama tanımlama, geliştirme, test, dökümantasyon ve bakım için ilgili alanların doldurulması yaklaşımını sağlamaktadır.
- 2- Uygulama tanımlamaya iletilerle yol gösterme yaklaşımı ve etkileşimli yazım kontrolü temin etmektedir.
- 3- Açıklamalı, anında yardım seti ve öğretici bilgiler, uygulama geliştirme süresince herhangi bir noktada yardımcı olmaktadır.

¹⁵³ Anonymous, Cross System..., a.g.e., s. 2

- 4- Deneme ekranları ve prototip çalışmaları programcılarla, uygulamanın kullanıcıları arasındaki iletişimi artırıp kolaylaştırmaktadır.
- 5- Komut arabirimi sayesinde, toplu iş modunda bakım çalışmalarının yapılması kolaylaşmaktadır.

CSP kısaca geleneksel programlama yaklaşımını yani ;

Kodlama → Derleme → Test Etme → Hata Ayıklama

şeklindeki lineer devreyi tartışmaya açmaktadır.

Uygulama kısmen bir kez kodlandığında program kütüphanelerinden doğrudan test edilebilmektedir. CSP/AD'nin test imkânı, yazım hatalarını derhal tanımakta ve mantık hatalarına çok çabuk nüfuz edebilmeyi sağlamaktadır. Tüm bir işlem (process) kesintisiz ve etkileşimli bir ortamda tamamlanabilmektedir.

CSP/AD bir uygulama tanımlama adımlarını -uygulamayı aksettirecek şekilde- üç seviyeye ayırmaktadır.

- 1-Veri tanımları : Bu aşamada veriye erişimi sağlayacak kayıtların (records), veri alanlarının ve dosyaların tanımlanması işlemleri yapılmaktadır.
- 2-Ekran tanımları : Bu adımda ise “map” olarak isimlendirilen etkileşimli ekranlar veya rapor formatları hazırlanmaktadır.
- 3-Mantık tanımları : Bu seviyede işlem bloklarının (ekran/yazıcı panelleri ile kayıt erişimleri sağlayan bölümlerin mantıksal akışı) tanımlanmaktadır¹⁵⁴.

5-Son kullanıcıların tatmini

CSP ürün setinin faydaları daha ziyade son kullanıcıları aşağıdaki yollarla tatmin etmektedir:

- 1- Detaylı mantık eklenmeden önce test ve prototip imkânları ile uygulama geliştirme süresince verimliliği artırmak,
- 2- Bir çok farklı işletim çevresinden bir diğerine uygulamaların taşınabilirliğini sağlamak,
- 3- Taşınabilirliğin yerleşik olmasından dolayı değişik işletim sistemleri arasındaki farklılığı gizlemek,
- 4- Baştan başa yapısal ve tutarlı bir yaklaşım kullanılarak uygulama ve bakım maliyetlerini azaltmak,
- 5- Yeni ve yüksek derecede yeterli uygulamalar geliştirmek için iş çevresinde çok görevli iş istasyonlarından istifade etmek,
- 6- Uygulamaları son kullanıcıların ihtiyacına uydurmak, bunun için de geliştirme zamanına en çabuk uyum sağlayacak kullanıcıları temin etmek,
- 7- Uygulama geliştirme safhasını azaltarak uygulamaya yapılacak desteği azaltmak,
- 8- Veriye ait karakteristikler bir kez tanımlanıp diğer uygulamalar ve programlar tarafından paylaşılabilirdiği için veri tanımlama devresini hızlandırmak,

¹⁵⁴ Anonymous, Cross System..., a.g.e., s. 2-4

9- Uygulama geliştirme adımlarında rehberlik edecek menüler, yardım ekranları ve çeşitli mesajlarla uygulamayı elverişli ve kolay kılmak,

10- İşlemlerin çoğunu otomatik hale getirerek (otomasyona bağlayarak) uygulama dökümanlarının maliyetini azaltmak.

2.9.2.1. CSP/AD Uygulamalarının Hazırlanması

CSP ürün seti ile yazılım geliştiren bir kişi, uygulamaların ortaya konması ve bakımı için mutlaka aşağıdaki basit adımları tatbik etmelidir:

1-Uygulamada kullanılacak bilgilerin tanımlanması

2-Panellerin düzenlenmesi

3-Uygulamanın tanımlanması

4-Uygulamanın test edilmesi

5-Uygulamanın derlenmesi

CSP/AD, programcıya program oluştururken menüler vasıtası ile yardımcı olan bir pakettir. Onlar da bu paneller vasıtası ile CSP/AD ve CSP/AE'yi öğrenerek bir uygulama veya ilgili uygulamaya ait gelişmiş kütüphanelerin yönetimini gerçekleştirebilmektedirler. Daha sonra uygulama tanımlanmakta, test aşamasından geçirilip derlenmektedir. Uygulama geliştirme, kullanılan sistemden bağımsızdır yani farklı sistemlerde de işletilebilmektedir. Böylece programcılarının verimliliği artmaktadır. Uygulamalar tanımlanırken üç tür ekran kullanılmaktadır.

1-Menüler : Menülerde programcılara çeşitli seçenekler sunulmaktadır. Uygulamamanın geliştirilmesinde ve değiştirilmesinde bu seçenekler kullanılmakta, kullanılan her seçenek yeni bir ekranı görüntülemektedir. Bu tür programlara menü-driven programlar denmektedir. Yani bir sonra gelecek olan ekran seçime bağlı olarak değişmektedir. CSP/AD ekranlarında seçenekler numaralı olup, her bir ekranda Pf1 veya Pf13 tuşuna basmak suretiyle o ekran ile ilgili bilgilere ulaşılabilir. Pf3 veya Pf15 tuşuna basılarak girilen veriler onaylanmış olmakta bundan sonra yeni bir ekran veya mesaj görüntülenmektedir.

2-Bilgi giriş ekranları : Giriş ekranları, uygulamanın tümü veya bir adımı hakkında gerekli tanıtıcı bilgileri vermek için kullanılmaktadır. Bilgi girilebilecek şekilde boş alanlar ekrana gelmekte, programcı tarafından doldurulmaktadır¹⁵⁵.

3-Serbest alanlar : Bu alanlar uygulama panellerini tanımlamak için kullanılan, şeklini tamamen uygulama geliştiricilerinin belirlediği ekranlardır.

2.9.2.2. CSP/AD'nin Yapısı

CSP/AD oldukça modüler bir programlama yapısı kullanmaktadır. Bu modüler yapı uygulama adımlarının ayrı ayrı üniteler olarak tanımlanmasından kaynaklanmaktadır. Böylelikle program tanımlarının değiştirilmesi ve tekrar tekrar kullanılması mümkün olmaktadır. Bir uygulamaya ait her tür tanım kullanıcının MSL (Member Specification

¹⁵⁵ Anonymous, Cross System..., a.g.e., s. 4-5

Library) denilen alanlarında saklanmaktadır. MSL'ler "member" denilen daha önceden tanımlanmış elemanlardan oluşmakta ve her elemanın kendine özgü bir adı bulunmaktadır. Bu elemanlar birçok uygulama programı tarafından ortaklaşa kullanılabilir. CSP/AD temel olarak yedi adet imkân (facility) sunmaktadır. Facility kelimesi "bir şeyi kolay yapmanın yolu" anlamına gelir ki CSP/AD'nin asıl gayesi de budur. Bu imkânlar şunlardır¹⁵⁶.

1-Liste işlemci (list processor) : Bu seçenekle MSL tanımlarını değiştirme, listeleme, silme, arama v.b. işlemler yapılmaktadır.

2-Tanımlama (definition) : Tanımlama seçeneğinde MSL elemanları tanımlanmaktadır. Bu elemanlar kayıtlar (records), bilgi parçaları (data items), tablolar (tables), paneller (maps), uygulamalar (applications), işlemler (process), deyim grupları (statement groups)'dır.

Bilgi tanımlarında, veritabanı veya kütük ile birlikte kullanılacak olan kaydın yapısı ve özellikleri tanımlanmaktadır. Bilginin tip ve uzunluğu için varsayılan değer üç uzunluklu karakter bilgisidir. Çalışma alanları (working storage)'da burada tanımlanmaktadır. Bu değişkenler sadece program çalışırken hafızaya yüklenmektedir. Genelde veriyi bir yerden diğerine aktarmadan önce düzenlemek ve elde tutmak amacıyla kullanılmaktadırlar. Ayrıca uygulama programına parametre aktarımını da sağlamaktadırlar.

Panel tanımlama, programlarda kullanılacak ekranların tanımlandığı bir bölümdür. Her panelin bir adı ve bir grubu vardır. Panelin kullanılacağı terminal tipi, panelden girilecek verilerin özellikleri, sadece belli değerlerin kabul edilmesi, veri girişlerinde meydana gelen hatalardan dolayı mesaj verilmesi gibi özelliklerin belirtimi bu bölümde yapılmaktadır.

CSP/AD'de işlem mantığı; panellerin ekrana getirilmesi, kütüklere ve veritabanına erişilmesi, kayıt, veritabanı ve paneller arasında veri aktarımı, aritmetik ve diğer işlemlerin yapılmasından ibarettir. Bu mantık yapısı, process adı verilen deyim gruplarında tanımlanmakta ve bir CSP/AD uygulamasının yapı taşlarını teşkil etmektedirler. Her bir işlem -Execute seçeneği hariç- tek bir I/O işlemini yerine getirmektedir. Bu I/O işlemine "process option" denmektedir. Bir paneli görüntülemek, bir kayıt okumak, veritabanına erişmek hep "process option" olarak belirlenen görevler olmaktadır. Dolayısı ile her bir işlemin bir nesnesi vardır (kayıt, panel, ...v.b.). Her bir I/O işlemi ile process deyimleri de işlenebilmektedir. Uygulama, process sıralarından oluşmakta ve "perform" deyimini ile çalıştırılmaktadır. Ana process'lerin işletim sırası yukarıdan aşağıya doğru olmakla beraber "flow" ifadeleri ile akış yönü belirlenebilmektedir.

Son olarak, uygulamaların kullanacakları bilgileri, veritabanından nasıl elde edecekleri de tanımlanmalıdır. CSP/AD hiyerarşik ve ilişkisel veritabanlarına erişip buradaki bilgileri de

¹⁵⁶ Anonymous, 1989, Cross System Product Reference, Ege Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi, s. 1-2

kullanabilmektedir. SQL/DS satırları, CSP/AD'nin kayıtları olmakta, uygulama geliştirirken SQL ifadeleri otomatik olarak üretilmektedir¹⁵⁷.

3-Test : Sağlanan kolaylıklardan biri de tamamen etkileşimli test imkânıdır. Uygulama tanımı kısmen tamamlanmış ise bu bölümler hemen test edilebilmektedir. Programın mantıksal işletimi sayesinde mantık kontrolleri de yapılabilmektedir. İzleme (trace) seçeneği aktif ise işlenen her deyim, kullanılan veriler ve sonuçlar ekranda gösterilmekte, böylece programın istenen yönde işleyip işlemediği izlenmekte, hatalar kolayca belirlenebilmektedir. CSP/AD uygulamanın işletimini test imkânı ile simule ederek çalıştırmaktadır¹⁵⁸.

4-Derleme (generation) : Uygulama test bölümünü başarıyla geçtikten sonra derleme işlemine hazır demektir. Bu işlemle üzerinde çalışacağı sistemde doğrudan çalışabilecek şekle getirilmiş olmaktadır.

5-Hizmetler ve dosya bakım imkânları (utilities and file maintenance facility) : Bu bölüm MSL veya uygulamaların yazıcı çıktılarının alınması, elemanların kopyalama, silme, isim değişikliği v.b. için gerekli hizmet programlarının bulunduğu yerdir.

6-Öğretici (tutorial) : CSP/AD'yi öğreten metin dosyalarına erişimi sağlayan bu bölüm belli konular hakkında başvuru kaynağı olarak da kullanılabilir.

7-MSL seçimi (MSL selection) : Program geliştiricilerin yaptıkları tanımlamaların nerede saklanacağı, bu seçenekte belirtilir. Belli MSL'ler ortak olarak kullanılabilir, aynı bilgilerin birçok yerde bulunması engellenmiş olmaktadır. Aynı anda en fazla altı adet MSL (kullanıcı MSL'i hariç) ile bağlantı kurulabilmektedir.

2.9.2.3. Uygulamaların Tanımlanması

CSP/AD, uygulama programlarının tanımlanmasında bakım kolaylığı, tanımların yeniden kullanılabilmesi ve modüler gelişim sağlayabilmek için ileri doğru yaklaşım metodunu kullanmaktadır. Tanımlama imkânları sayesinde, uygulama geliştiricilerin hazırladıkları veri tanımları, ekranlar ve uygulama mantığının diğer bir uygulamadan bağımsız olmaktadır¹⁵⁹.

1-Veri tanımlama : PL/1 veya Cobol dilindeki veri bildirimine benzer şekilde uygulama programının kullanılacağı verilerin hususiyetlerinin, özelliklerinin açıklanması ve tanıtılması gerekmektedir. CSP/AD programının veri elemanlarına ait tanımları, kütüphanesinde ayrı üyeler olarak ve veriye ait özellikler diğer elemanların tanımı ile birlikte -bu tanımlar paylaşıldığından otürü- bağımsız olarak saklanmaktadır. Her zaman kullanılan veri elemanlarına ait özellikler uygulamalarda bir standardizasyon sağlayıp hata oluşumunu en aza indirilmeyi sağlamaktadır. Veri tanımlama; veri elemanlarının, kayıtların ve tabloların tanımlanmasından ibarettir. Herhangi bir dosya veya veritabanında

¹⁵⁷ Anonymous, Cross System..., a.g.e., s. 11

¹⁵⁸ Anonymous, Cross System..., a.g.e., s. 3-9

¹⁵⁹ Anonymous, Cross System..., a.g.e., s. 5

mevcut veri elemanının biçimi, kayıt tanımının tipini belirtmektedir. Kayıtlar ve tablolar, uygulama geliştirenlerce yerel veya genel olarak tanımladıkları veri parçalarından oluşmaktadır. Yerel olarak tanımlanan bir veri elemanı uygulama içerisinde yer alan diğer kayıt veya tablolarca ortak olarak kullanılmaz ve paylaşılamaz. Fakat veri elemanı genel olarak tanımlandığı zaman uygulamada yer alan diğer kayıt veya tablolar tarafından ortaklaşa kullanılabilir. CSP/AD uygulamaları sıralı, ilişkisel, indexli kütüklere, DL/I veya ilişkisel bir veritabanındaki kayıtlara erişebilmektedir. Tablolardaki veri elemanları ise uygulama çalışırken referans noktaları veya düzenleme amacıyla kullanılmaktadır.

2-Ekran (map) tanımlama : Cobol veya PL/I gibi dillerle geleneksel uygulama programı geliştirirken uygulama geliştirme çabasının önemli bir bölümünü ekranların düzenlenmesi ve biçimlendirilmesi çabaları teşkil etmektedir. CSP/AD bu çabaları, -yapısal programlamada olduğu gibi ihtiyaç duyulan tekrarlamaları kaldırarak ve ekran planları sağlayarak- mesajlarla yönlendirme yaklaşımıyla ele almaktadır. Ekran tanımlama süresince, uygulama geliştirenler bir uygulama için kullanıcı ekranları ve yazıcı raporları düzenlemekte, tanımlanan bu ekranlar bir uygulamanın çalışması süresince girilecek veya görüntülenecek veri planını tanımlamaktadır. Her bir ekran bir ekran grubuna (map group) ait olup tek bir isme sahiptir. Ekran grubu bir uygulama programının kullandığı ekranların tümüdür. CSP/AD bir uygulama programı için ekran planlarının tanımlanmasına, giriş/çıkış düzenlemelerine, cihaz (device) özelliklerinin belirtilebilmesini sağlamaktadır. Ekran tanımlamalarının bir diğer önemli özelliği uygulama geliştirenlere, bir uygulamada tam olarak hangi ekranı görüntüleyeceğine imkân tanımasıdır. Ekran tasarımlarında oldukça serbest (free-form) görüntüler sunulmaktadır. Uygulama geliştirici, uygulama çalıştığı sürece imleç (cursor) veya özel kodlar kullanarak sabit ve değişken sahaların tam olarak nerelerde görüntüleneceğini kararlaştırabilmektedir. Değişken sahası, uygulama çalıştığında kullanıcı tarafından veya uygulama mantığınca verinin girilebildiği ve değiştirilebildiği ekran üzerindeki alanlara verilen isimdir. CSP/AD, ekran üzerindeki her bir değişken sahanın özelliklerini gösterme, düzeltme ve doğru tanımların yapılabilmesini sağlamaktadır. Örneğin, programcı bir CSP/AD panelinde uygun bir mesaj kullanarak verinin ne türde girilebileceğini veya görüntülenebileceğini söyleyebilmektedir¹⁶⁰. Özet olarak uygulama geliştirenler her bir sahaya ait cihaz özelliklerini -renk, ışıklandırma, imleç pozisyonu, alanların parlaklığı v.b.- tanımlayabilmektedirler.

3-Akış mantığını tanımlama : Veri alanları tanımlanıp raporlar ve planlar düzenlenince, programcı mantık akışı ile beraber bunları birleştirmeye çalışır. Uygulama mantığının yönü, yukarıdan aşağıya doğru olup işlemler veya ifade grupları (statement group) olarak isimlendirilen mantıklı bloklardan oluşmaktadır. Her bir işlem grubu -bir kaydın okunması veya yazılması, terminalde bir panelin görüntülenmesi v.b. gibi- tek bir giriş/çıkış faaliyetini yerine getirmektedir. Örneğin, bir uygulama bir envanter dosyası üzerinde düzeltme, ekleme ve bakım işlemlerini yürütüyorsa aşağıdaki adımların uygulanmasını gerektirmektedir.

¹⁶⁰ Anonymous, Cross System..., a.g.e., s. 5-6

1. adımda kullanıcı, üzerinde değişiklik yapmak üzere bir envanter kaydını ekranları kullanarak seçilmeli, 2. adımda envanter kaydı ilgili dosya veya veritabanından okunmalı, 3. Adımda kayıt üzerindeki bilgi kullanıcının uygun değişiklikleri yapabilmesi için görüntülenmeli ve 4. adımda kayıt üzerinde yapılan değişiklikler veritabanına yeniden aktararak başka bir kayıt seçilmek üzere ilk adıma dönmelidir.

Yukarıdaki adımların her biri CSP/AD'de işlem olarak tanımlanmaktadır. İşlemler, bir uygulamanın yapı taşlarıdır. Uygulamanın icra edeceği tek bir görev veya fonksiyona ise "process" denilmektedir. Her bir işlemin diğer uygulamalar tarafından kullanılmasını sağlayan tek bir ismi bulunmaktadır. Bir CSP/AD uygulaması, çeşitli işlem bloklarından oluşmakta ve işlem blokları mantıksal ifadelerden, ekran, yazıcı veya veri dosyaları üzerinde giriş-çıkış faaliyetlerini icra eden işlem tercihlerinden, sadece mantıksal ifadeleri işleyip giriş/çıkış işlemleri yapmayan ifade gruplarından ve özel fonksiyon kelimelerinden oluşmaktadır.

2.9.2.4. İşlem İfadeleri

İşlem ifadeleri aritmetiksel işlemler, verilerin taşınması, tabloların kullanımı, ekran sahalarına ve veri kayıt parçalarına erişim gibi fonksiyonları ile yüksek seviyeli bir dile destek sağlamakta, bir işlemin öncesinde veya sonrasında çalıştırılabilmektedir. CSP/AD'de çeşitli işlem ifade tipleri vardır ve bunlar aşağıda verilmiştir¹⁶¹.

1.Aritmetiksel ifadeler : Bu ifadeler bilinen toplama, çıkarma, çarpma, bölme ve kalan bulma gibi ifadelerdir

2. Veri ifadeleri

MOVE	Verileri; veri parçalarına, ekranlara, kayıtlara, tablolara ve depolama alanları (working storage) kısmına taşır.
MOVEA	Tek bir veri parçasını, bir diziye veya bir diziyi diğer bir diziye aktarır.
RETRIEVE	Bir tablodan veri elde etmek için kullanılır.
SET	Ekranlar, kayıtlar veri parçaları için özel şartları düzenler.

3.Şart ifadeleri

FIND	Bir tablodaki veriyi arar. Veri bulunursa ilgili işlem veya ifade bloku, veri bulunmаса bu durumda muhtemelen belirtilmiş olan diğer işlem veya ifade grubu çalışır.
IF/ELSE / END	Karşılaştırma esasına dayanan ve bir şarta bağlı işlem ifadeleridir.
WHILE / END	İşlem ifadelerinin şarta bağlı olarak tekrarlanmasını sağlar.
TEST	Özünde ekran ve kayıtları test etmek olup şarta bağlı olarak ifade grupları veya işlemleri çalıştırır.

¹⁶¹ Anonymous, Cross System..., a.g.e., s. 6-7

4.Uygulamaları irtibatlandıran ifadeler

CALL	CSP türünden olan veya olmayan bir programı, bu programın kayıtlarını, ekranlarını hafıza değişkenlerini, veri parçalarını ve karakter ifadelerini içerikleri ile beraber seçilen programa aktararak programın çalışmasını başlatır.
XFER/ DXFER	Başlangıç uygulamasına dönüş yapılmaksızın diğer bir uygulamaya transfer sağlar. Kayıtların içeriği isteğe bağlı olarak, çalışacak programa aktarılabilir. Ekran panelleri ise XFER ifadesi ile isteğe bağlı olarak aktarılabilir.

5.Uygulama yapısını ilgilendiren ifadeler

PERFORM	Bir işlem içerisinde yer alan bir diğer işlem veya ifade grubunu çalıştırır. Bu şekilde ilişkili fonksiyonları hiyerarşik bir düzene sokulmuş olur.
---------	---

6.İfade grupları : Bu program grupları isimleri belirtilerek çalıştırılmaktadır. İfade grupları, uygulamaların modüler bir şekilde meydana getirilmesinden dolayı diğer bir işlem veya ifade grubu içerisinde çalıştırılabilmektedir. Böylece programcılar sık sık kullanılan fonksiyonlar için yeniden kodlama yapma gereği duymayacaktır. Çalışma bitince uygulama kontrolü başlangıç noktasına bırakılmaktadır.

2.9.2.5. İşlem Çeşitleri

CSP -execute işlem seçeneği hariç- her bir işlem seçeneği, ismi kayıt veya ekran olan nesneleri- yönetmektedir. CSP/AD'de kullanılan işlem çeşitleri aşağıda verilmiştir¹⁶³ :

CONVERSE	Bir ekranı görüntüler ve kullanıcı tarafından girilen verileri daha önceden yapılan tanımları kullanılarak kontrol yapan işlemlerdir.
DISPLAY	Uygulama verilerin düzenlenmesi, görüntülenmesi veya bir ekranın yazıcıdan alınması işlemlerini yapar.
INQUIRY	Bir dosya veya veritabanından bir kaydın sorgulanma amacı ile ele alınması işlemidir.
UPDATE	Bir dosya veya veritabanından bir kaydın değişiklik veya silme amacıyla ele alınması işlemidir.
REPLACE	Bir dosya veya veritabanına üzerinde değişiklik yapılan kaydın yeniden aktarılması işlemidir.
ADD	Bir dosya veya veritabanına yeni bir kaydın yerleştirilmesidir.
DELETE	Bir dosya veya veritabanında mevcut bir kaydın silinmesi, çıkarılması işlemidir.
SETINQ	Bir ilişkisel veritabanındaki satırları seçilen bir şarta bağlı olarak düzenleme işlemidir.
SCAN	Bir sıralı dosya veya veritabanındaki bir sonraki kaydın ele alınmasıdır.
SCANBACK	Bir sıralı dosya veya veritabanındaki bir önceki kaydın ele alınmasıdır.
CLOSE	İlişkisel bir veritabanının taranmasını bitirmede, yazıcıların bağlantısını kesmede ve bir dosyayı kapatmada kullanılan işlemidir.
SETUPD	Değişiklik veya düzeltme amacı ile bir ilişkisel veritabanındaki satırların seçilen bir şarta bağlı olarak düzenlenmesidir.
EXECUTE	Herhangi bir terminal veya giriş-çıkış operasyonu içermeyen başlangıç parametrelerini içeren işlem ifadeleridir.
SQLEXEC	İlişkisel bir veritabanında kullanıcı tarafından tanımlanan SQL ifadelerini çalıştıran işlemlerdir.

¹⁶³ Anonymous, Cross System..., a.g.e., s. 8

CSP/AD'de "EZE" ön eki ile başlayan kelimeler uygulamalar için özel fonksiyonları -hata mesajlarının görüntülenmesi giriş-çıkış operasyonlarının kontrolü, servis rutinlerinin çağırılması, servis kontrol ayarlarının düzenlenmesi v.b. - tanımlamaktadır.

2.9.2.6. MSL Paylaşımı

Uygulama geliştiriciler, dosyalar ve veritabanı tanımlarını saklayan MSL elemanlarını paylaşmak zorundadırlar. Dosya ve veritabanları tanımlanıp oluşturulduğunda uygulama geliştirici bunları sadece okunabilen genel bir MSL içine yerleştirmelidir. Özel uygulama geliştirenler bu tanımlara MSL'in dizilim özelliğinden faydalanarak erişmektedirler. Dizilim özelliği, MSL'i bir set olarak düzenlemekte ve uygulama geliştirenlere bir kaynak kütüphane olarak görünmektedir.

2.9.2.7. ALF (Application Load File)

Yapılan tanımlar, CSP/AE ile çalıştırılmak üzere derlenerek ALF olarak isimlendirilen özel bir kütüphane veya katalog içine aktarılmaktadır. CSP/AE, ALF'den derlenmiş haldeki tanımları yorumlayarak almakta ve çalıştırmaktadır. ALF uygulaması CSP/AD kullanılarak geliştirilen bir uygulama setidir ve CSP/AE tarafından kullanılan derlenmiş yükleme modüllerinin yönetimini temin etmektedir¹⁶⁴.

2.9.2.8. Harici Kaynak Biçimli Dosyalar

CSP seti, CSP/AD, CSP/AD programlanabilir Terminaller ve diğer dizayn araçları ile harici kaynak biçimli dosyalar kullanarak, CSP/AD ile geliştirilen uygulamaların CSP çevresine ithal edilmelerini sağlamaktadır. Harici kaynak biçimi, uygulamanın değişik elemanları olarak tanımlanan ve etiketler (tags) olarak isimlendirilen bir anahtar kelime setidir. CSP dışındaki bir uygulama "import" uygulaması kullanılarak harici kaynak biçimi etiketlerine çevrilmektedir. CSP çevresindeki uygulamalar ise "export" uygulaması ile harici kaynak biçimlerine dönüştürülür. "Export" uygulaması ile programcı MSL tanımlarını harici kaynak biçimli dosyalara kopyalayabilmekte, test ve derleme amacıyla ana uygulamaya taşıyabilir.

CSP program seti veritabanı desteği, geliştirme kütüphaneleri, toplu veya etkileşimli uygulamalar, geniş sistem desteği, IBM PC, IBM programlanabilir terminalleri ve AS/400 desteği, Cobol programlarını derleme, diğer programlama dillerine arabirim, ANSI SQL desteği ve ulusal dil desteği de sunmaktadır¹⁶⁵.

2.9.3. Sistem Analizi

Bu çalışmamızda konuya Sistem Analizi ile yaklaşılmaya çalışılmıştır. "Sistem bir veya daha çok amaca veya sonuca ulaşmak üzere aralarında ilişkiler olan fiziksel veya kavramsal, birden çok bileşenin oluşturduğu bütündür. Araştırmacılar kendi problemlerine özgü bir sistem kavramı benimsemekte ve çalışmalarında bu kavramı bütün sistemleri kapsar bir şekilde kullanmaktadırlar. Örneğin bir matematikçi için sistem sözcüğü değişkenler ve bağıntılar takımı olurken, işletmeci için işletme çevresi ve ararlarındaki ilişkiler dinamiğidir. Yine bir bilgi işlemci için sistem elektronik bilgi işlem

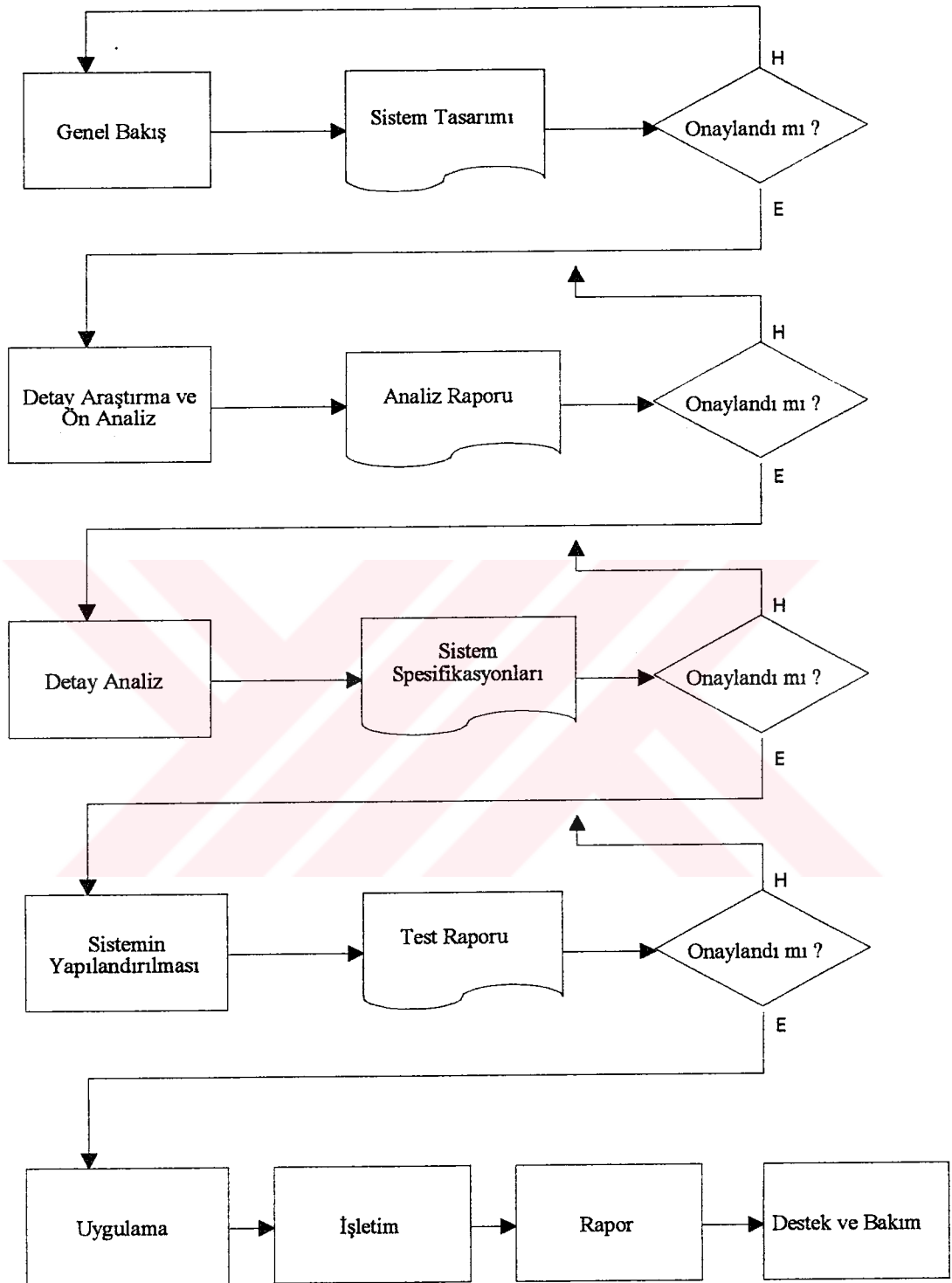
¹⁶⁴ Anonymous, Cross System..., a.g.e., s. 12

¹⁶⁵ Anonymous, Cross System..., a.g.e., s. 13

araçları ve yazılımı olmaktadır¹⁶⁶.” Herhangi bir şeyi en ufak parçalarına, ayrıntılarına ayırmaya ise analiz denmektedir. Sistem analizi ise mevcut sistemin daha iyi bir sistem tasarlama ve geliştirmeye temel oluşturmak üzere tüm yönleriyle analiz edilmesi diye tanımlanmaktadır. Bilgisayar için sistem analizi dendiğinde ise yukarıda belirtilen "daha iyi bir sistem tasarlama ve geliştirmeye temel oluşturmak üzere" cümlesi "sistemi bilgisayara aktarabilecek şekilde tasarlamak üzere" diye değiştirilebilir. İyi bir sistemin kurulabilmesi için Şekil 14’de belirtilen adımlar doğrultusunda hareket etmek gerekir¹⁶⁷.

¹⁶⁶ Emin D. Aydın, Ekim 1991, Bilgi Bilimi İnnformatik -Genel Sistemler- Sibernatik ve Kitle İletişimi, Aydın özel Eğitim ve Yayıncılık, İstanbul, s. 150

¹⁶⁷ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 263-264



Şekil 14. Sistem Analizinin Adımları

Bu adımların içeriklerini izah edecek olursak :

1-Genel bakış : Bu bölümde konu belirtilmekte, amaçlar doğrultusunda sistem ele alınıp incelenmektedir.

2-Sistem tasarımı : Genel bakış adımı sonrası çıkacak rapordur. Sistemin amaçları, sonuca yaklaşım, sistemin yararları, proje grubu ve zaman süreci adımlarından oluşmaktadır.

3-Detay araştırma ve ön analiz : Bu aşamada iyi bir sistem oluşturmak için uyulması gereken kurallar ışığında sistem detaylı bir şekilde analize tabi tutulmaktadır.

4-Analiz raporu : Üçüncü adım sonrası elde edilecek olan rapor olup amaçların ve sistemin yararlarının detaylı olarak tekrarı, sistemin temeli (açıklamalar ve sistemin akış diyagramı), çıktılar (raporlar, raporların düzeni ve sıklığı), girdi tipleri ve yoğunlukları, sistemin işletimi ve uyarılma teklifi adımlarından oluşmaktadır.

5-Detay analiz : Sistem için gerekli olan bütün özellikler en ince noktasına kadar bu safhada analiz edilmektedir.

6-Sistem spesifikasyonları : Beşinci adım sonrası ortaya çıkan rapordur. Sistemin genel tanımı, sistemde kullanılan kodlama sisteminin açıklanması, bilgi giriş formlarının detaylı olarak hazırlanması, akış diyagramlarının hazırlanması, kontrollerin listesi, rapor çıktı formlarının detaylı olarak hazırlanması, bilgilerin bilgisayarda depolanacakları ve işlenecekleri dosyaların özelliklerinin belirtilmesi, sistemin bilgisayara uyarlanması için gerekli olan bilgisayar programlarının hazırlanması, sistemle ilgili eklerin dökümü bu adımda yerine getirilmektedir.

7-Sistemin inşası : Sistem spesifikasyonları onaylandıktan sonra sistemin çalışabilmesi için gerekli tüm aktiviteler (programlama, kodlama, test v.b.) bu adımda yapılmaktadır.

8-Test raporu : Sistemin inşası sonucunda yapılan testler bir rapor haline getirilmekte ve örnekler onay makamına sunulmaktadır.

9-Uygulama : Test raporu onaylandıktan sonra uygulama için gerekli çalışmalara başlanmakta, bilgisayarlarının montajı, dökümanların basımı, personelin eğitimi, dosyaların açılması v.b. işlemler yerine getirilmektedir.

10-İşletim : Uygulama tamamlandıktan sonra sistem işletilmiş olmakta ve sorumluluk kullanıcıya bırakılmaktadır.

11-Rapor : Sistem hakkında genel bir rapor verilip amaçlarına erişilip erişilmediği tartışılmaktadır.

12-Destek ve bakım : Sistemin devamlılığı için çok gerekli bir adımdır. Çünkü her an için sisteme yeni ilaveler yapılabilir¹⁶⁸.

2.10. Kütüphane Otomasyonu Uygulama Programı

Atatürk Üniversitesi Bilgi İşlem Merkezi'nce hazırlanan Kütüphane Otomasyonu Uygulama Programı iki kısımdan oluşmaktadır. İlki kütüphanenin idari servislerine ait işlemleri yapan KU00A, diğeri yayın tarama işlemlerini yürüten TA10A programıdır. Uygulama programlarının çalışma şekillerine girmeden önce kullanıcı ile etkileşimi sağlayan paneller (map) hakkında bilgi vermenin faydalı olacağı kanaatindeyiz. Bir program paneli 3 bölümden meydana gelmektedir.

¹⁶⁸ Anonymous, Bilgisayar Ansiklopedisi, a.g.e., s. 265-267

1-Başlık bilgileri

2-Gövde kısmı

3-O ekranda kullanılacak fonksiyon tuşları kısmı

1-Başlık bilgileri : Ekranların bu bölümünde, en solda tasarlanan ekranın adı, hemen yanında o anda bu programı kullanan servis veya yetkili kullanıcı adı, orta kısmın üçüncü satırında bu panelin hangi servise hizmet verdiği, ekranın sağında ise tarih bilgisi yer almaktadır. Seçeneksiz ekranlarda ayrıca sahife v.b. bilgiler yine ekranın sağ kısmında yer almaktadır. Örnek bir başlık kısmı aşağıda verilmiştir.

KU20M02-KATALOG	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Süreli Yayınlar Servisi	12 /08/1997
-----------------	---	-------------

2-Gövde kısmı : Gövde kısmı, ekranların seçenekli veya bilgi giriş ekranı olup olmamasına göre değişiklikler arz etmektedir.

a-Ekranda seçenekler varsa görüntü aşağıdaki gibi olmaktadır.

1- Temel Kart (Yazar Adı) 2- Kitap Adı 3- Konu Katalogu 4- Dizi Katalogu 5- Yer Katalogu 6- Çıkış Tercihinizi Giriniz : 2

b-Ekran bilgi giriş alanlarına sahipse, örnek bir görüntüsü aşağıdaki gibi olmaktadır.

Seri (DİZİ) Adı :	
Konu Başlıkları :	
Bibliyografya :	
Açıklamalar :	
Dizin Bilgisi :	
Bakı Kaydı :	
Sayfa ve Cilt :	
Fiziki Özellik. :	
Yayın Dili :	Kayıt Ortamı :
Bu Yayına Daha Önceden Verilmiş Demirbaş Numaraları :	
İzleme Kaydı :	

3-Tuşlar ve fonksiyonları : Bu kısım da yine ekranların durumuna göre değişmektedir. En alt satır kullanıcıyı yönlendirici mesajlar için ayrılmıştır. Ekranda görüntülenecek olan her

tür mesaj bu alana aktarılmaktadır. Kullanıcı her bir ekranda PF1 tuşuna basarak yapacağı işlemler hakkında bilgi alabileceği ekranlara ulaşabilmektedir.

a-Ekranda seçenekler görüntüleniyorsa sadece tek bir tuş tanımlanmıştır. Örnek ekran aşağıda gösterilmiştir.

Enter : İşlem

b-Bilgi giriş ekranlarında, çeşitli tuşlar ve görevleri ise aşağıdaki gibi görünmektedir.

PF3 :Çıkış	PF4 : Ayrıntılar	PF7 : Geri	PF8 : İleri	PF9 :Yayın Takibi
------------	------------------	------------	-------------	-------------------

2.10.1. KU00A Uygulama Programı

KU00A programı, kütüphane işlemlerini yapmak üzere ilk çağrılan program olup ana menü durumundadır. Program çalıştırıldığı zaman KIMLIK CSP tablosunda isimleri belirtilmiş olan yetkili kullanıcılar kontrol edilmektedir. Kullanıcıya bu tabloda yetki verilmiş ise KU00M01 (Ek 4.1.) paneli görüntülenmektedir. Bu ekranda 6 adet seçenek bulunmaktadır. Yanlış veya geçersiz bir veri değeri girilmişse kullanıcı çeşitli mesajlarla uyarılmaktadır. Yetkisiz kullanıcılar ise programa dahil edilmemektedir. Yetkili bir kullanıcının yapacağı seçimler, APPT01 CSP tablosundan kontrol edilmekte ve program akışı bu tabloda belirtilen programa geçirilmektedir. Working storage olarak KU00W kullanılmıştır. Bu programın yapısı Ek 3.1.'de, akış şeması Ek 5.2.'de verilmiştir. Ana menüden yapılacak seçimlere göre aşağıdaki akış planı uygulanmaktadır.

1-Ana menünün 1. tercihinin seçilmesi halinde, KU01A programı çalışmakta ve KU01M01 (Ek 4.2.) paneli görüntülenmektedir. Bu bölüm katalog ve sınıflandırma servisinin yapabileceği işlemleri göstermektedir. Yapılan seçime göre hangi program bölümünün devreye gireceği ise APPT02 CSP tablosundan kontrol edilmektedir. Working storage olarak KU01W kullanılmış, programın yapısı Ek 3.2.'de, akış şeması ise Ek 5.3.'de verilmiştir.

2-Ana menünün 2 nolu tercihinin seçilmesi halinde KU02A programı çalışmakta ve KU02M01 (Ek 4.3.) paneli görüntülenmektedir. Süreli yayın servisinin yapabileceği işlemleri gösteren bu ekrandan yapılacak ekran seçimler APPT03 CSP tablosundan kontrol edilmekte ve ilgili program devreye girmektedir. Bu programın yapısı Ek 3.3.'de, akış şeması Ek 5.4.'de verilmiştir. Working storage olarak KU02W kullanılmıştır.

3-Ana menünün 3. tercih seçilirse KU03A programı çalışmakta ve KU03M01 (Ek 4.4.) paneli vasıtasıyla sağlama servisi tarafından yapılacak işlemler görüntülenmektedir. Yapılan seçimler APPT04 CSP tablosundan kontrol edilip ilgili program çağrılmaktadır. Kullanılan working storage KU03W'dir. Programın yapısı Ek 3.4.'de, akış şeması Ek 5.5.'de verilmiştir.

4-Ana menü tercihi 4 ise üyelik ve ödünç verme servisinin yapabileceği işlemler KU04A programı vasıtasıyla ekrana getirilmektedir. Ekran panelinin adı KU04M01 (Ek 4.5.) olup yapılacak tercihler APPT05 CSP tablosundan kontrol edilecek ve ilgili program çağrılacaktır. Bu programın yapısı Ek 3.5.'de, akış şeması Ek 5.6.'da verilmiştir. Kullanılan working storage KU04W'dir.

5-Ana menü tercihinin 5 olması durumunda, KU05A programı dökümantasyon servisi hizmetlerini, KU05M01 (Ek 4.6.) paneli vasıtasıyla görüntüleyecektir. Yapılacak seçimler APPT06 CSP tablosundan kontrol edilip program akışı bu noktadan devam edecektir. Working storage olarak KU05W kullanılmıştır. Programın yapısı Ek 3.6.'da, akış şeması Ek 5.7.'de verilmiştir.

6-KU00A uygulamasının 6. seçeneği kullanıcının programı bitirip CMS ortamına dönmesini sağlamaktadır.

2.10.2. KU01A Uygulama Programı

KU01A programı katalog ve sınıflandırma servisi tarafından yapılacak işlemleri çalıştırmaktadır. KU01M01 panelinden yapılacak tercihlere göre işlem yapmaktadır.

1-KU01M01 panelinden yapılacak tercih 1 ise KU10A programı çalışmaktadır. Programın amacı; kitap, tez, makale bilgilerinin veritabanına girilmesini sağlamaktır. KU10A programının yapısı Ek 3.7'de, akış şeması ise 5.8'de verilmiştir. İlk ekran KU10M01 (Ek 4.7.) olup PF8 tuşu ile KU10M02 (Ek 4.8.) paneline geçilmektedir. Bilgi girişi tamamlandıktan sonra Enter tuşuna basılarak kayıt işlemi yapılmaktadır. Herhangi bir hata oluşmamış ise girilen bilgiler Kitaplar SQL tablosuna eklenmekte, hata yapılmışsa kullanıcı uyarılmaktadır. Kullanıcının, özellikle kodlanması gereken alanlarda hata yapmasının önüne geçmek için KU10H01 (Ek 4.9.) ve KU10H02 (Ek 4.10.) panelleri - PF1 tuşu ile erişip doğru bilgileri öğrenmesi için- düzenlenmiştir. Working storage olarak KU10W, record (kayıt) olarak KU10R01 kullanılmıştır. Bu ve diğer programlarla birlikte kullanılan CSP tabloları için Ek 1.11.'e, Kitaplar SQL tablosunun yapısı için Ek 2.1.'e bakınız.

2-KU01M01'den yapılan seçim 2 ise KU11A programı çalışmaktadır. Amaç, daha önceden girilmiş kitap, tez, makale bilgileri üzerinde değişiklik veya silme yapabilmektir. Yapısı Ek 3.8.'de akış şeması 5.9.'da verilmiştir. Program çalıştırıldığında ise KU11M01 (Ek 4.11.) paneli görüntülenmektedir. Bu panelin akabinde yapılacak tercihlere göre KU11M02 (Ek 4.12.) ekranından gerekli bilgi girilip Enter tuşuna basılmakta, bilgi bulunmuşsa KU11M03 (Ek 4.13.) paneli görüntülenmektedir. Gelen ekranda PF8 tuşu ile ileri, PF7 tuşu ile geriye doğru hareket edilmekte, kullanıcının herhangi bir bilgi satırı üzerinde iken PF4 tuşuna basması halinde KU11M04 (Ek 4.14.) paneline gidilmektedir. Bu panel daha önceden girilmiş bilgileri ekranada görüntüleyerek PF5 tuşu ile değişiklik, PF6 tuşu ile silme yapılabilmesini sağlamaktadır. PF7 tuşu KU11M05 (Ek 4.15.) paneline, yani diğer bilgilere geçilmektedir. PF1 tuşu ile yardım bilgileri ekranı KU11H01 (Ek 4.16.) görüntülenmektedir. Record olarak KU11R01, working storage olarak yine KU10W kullanılmıştır.

3-KU01M01 panelinden yapılan seçim 3 ise KU12A programı çalışmakta ve KU12M01 (Ek 4.17.) paneli ekrana gelmektedir. Programın yapısı Ek 3.9.'da, akış şeması ise 5.10'da verilmiştir. KU12M01 panelinden yapılacak seçilen her türlü tercihte ise KU13A programı çalışmaktadır. Program, KU12A'dan yapılan seçime göre veritabanından elde edilecek bilgilerin yazıcıdan alınmasını sağlamaktadır. KU13A'nın akış şeması Ek 5.11.'de, yapısı ise Ek 3.10.'da verilmiştir. Görüntülenen KU13M01 (Ek 4.18.) panelinin ardından, bu panelden girilen bilgilere bağlı olarak KU13M02 (Ek 4.19.) paneline gelmektedir. PF5 tuşuna basılarak gelen bilgilere ait katalog kartları yazdırılmakta, PF2 tuşu kataloglama seçeneklerini yeniden görüntülemekte ve yeni işlemlerin yapılmasına imkan vermektedir. PF8 tuşu ile ileri, PF7 tuşu ile geriye doğru gidilmektedir. Kullanılan record KU13R01, working storage ise KU12W adını taşımaktadır. Katalog kartları çıktı örnekleri Ek 6.14., 6.15., 6.16., 6.17., 6.18.'de verilmiştir.

4-KU01M01 panelinde iken 4. tercih seçilirse KU00A uygulamasına dönülecektir.

2.10.3. KU02A Uygulama Programı

KU02A programı süreli yayınlar servisinin yapacağı işlemleri içermektedir. KU02M01 panelinden yapılacak tercihlere göre işlem yapmaktadır.

1-KU02M01 panelinden yapılan tercih 1 ise süreli yayınlara ait eksik sayıların takibini yapan bir uygulama olan KU20A çalışmaktadır. Yapısı Ek 3.11.'de, akış şeması Ek 5.12'de verilmiştir. Ekrana KU20M01 (Ek 4.20.) paneli gelmekte ve çeşitli seçenekler içermektedir. Enter tuşuna basılarak KU20M02 (Ek 4.21.) paneline gelmekte, gerekli bilgiler yine Enter tuşu ile taranabilmektedir. Taranan bilgiler KU20R01 vasıtasıyla KU20M03 (Ek 4.22.) paneline taşınmaktadır. Gelen bilgiler üzerinde PF4 tuşuna basılarak herhangi bir yayının ayrıntıları KU20M04 (Ek 4.23.) panelinde sunulmakta, PF9 tuşu ile yayın takibi bölümüne geçilmektedir. Eğer süreli yayın çıkış aralığı günlük, düzensiz veya haftalık ise KU20M05 (Ek 4.24.) görüntülenmektedir. Bu panelin diğer kısmı olan KU20M06 (Ek 4.25.) paneline ise PF10 tuşu ile geçilmektedir. Yayının çıkış aralığı aylık ise KU20M07 (Ek 4.26) paneli ekrana gelmektedir. Veri girişi tamamlanınca Enter tuşuna basılarak bilgiler Eksik_sayı SQL tablosuna KU20R02 yardımıyla aktarılmaktadır. Eksik_sayı SQL tablosunun yapısı Ek 2.4.'de verilmiştir. PF9 tuşu ile o yayına ait eksik sayı bilgileri yazıcıdan alınabilmektedir. PF1 tuşu ile ilgili yardım bilgilerine, yani KU20H01 (Ek 4.27) ve KU20H02 (Ek 4.28) panellerine erişilebilmektedir. Working storage olarak KU20W ve KU20W01 kullanılmıştır. Süreli yayın çıkış aralıkları Ek 1.4'deki KU11T04 CSP tablosunda verilmiştir. Çıkış aralığı düzensiz yayınlara ait çıktı örneği Ek 6.12.'de, diğerlerine ait yazıcı çıktısı ise Ek 6.13.'de gösterilmiştir.

2-KU02M01 panelinden yapılan seçim 2 ise KU21A programı çalışmaktadır. Amacı süreli yayın servisi katalog listesinin yazıcıdan alınmasını sağlamaktır. Akış şeması Ek 5.13.'de, uygulama yapısı Ek 3.12.'de verilmiştir. İlk olarak KU21M01 (Ek 4.29.) ekranı görüntülenmektedir. Yapılacak işlemler ekran üzerinde gösterilmiştir. Program çalıştırıldığı zaman bilgiler doğrudan yazıcıya gönderilmektedir. PF1 tuşuna basılarak KU21H01 (Ek 4.30) yardım paneline ulaşılmaktadır. Record olarak KU21R01 ve

KU21R02, working storage olarak KU21W kullanılmıştır. Ek 6.19.'da katalog listesine ait çıktının şekli yer almaktadır.

3-KU02M01 panelinden 3. tercih seçilmişse KU22A programı çalışmaktadır. Amacı süreli yayın servisi katalog kartlarının yazıcıdan alınmasını sağlamaktır. Akış şeması Ek 5.14.'de, uygulama yapısı Ek 3.13.'de verilmiştir. Öncelikle KU22M01 (Ek 4.31.) ekranı görüntülenmekte, gerekli bilgilerin girilip Enter tuşuna basılması halinde ilgili katalog kartları yazıcıdan alınmaktadır. PF1 tuşuna basılarak KU22H01 (Ek 4.32) yardım paneline ulaşılmaktadır. Record olarak KU22R01, working storage olarak KU22W ve KU22W1 bu KU22A programında kullanılmıştır.

4-Yapılan tercih 4 ise KU00A uygulamasına dönülmektedir.

2.10.4. KU03A Uygulama Programı

KU03A programı sağlama servisinin yapacağı işlemleri çalıştıran bir programdır. KU03M01 panelinden yapılacak tercihlere göre işlem yapmaktadır.

1-KU03M01 panelinden yapılan seçim 1 ise KU30A programı çalışmaktadır. Akış şeması Ek 5.15.'de, yapısı ise Ek 3.14.'de verilmiştir. Amacı yapılacak sipariş bilgilerini Kayıt SQL tablosuna kaydetmektir. Kayıt SQL tablosunun yapısı için Ek 2.3.'e bakınız. Ekranı gelen KU30M01 (Ek 4.33.) panelinden yapılacak tercihlere göre farklı paneller görüntülenmektedir. Kitap siparişleri için KU30M02 (Ek 4.34.), süreli yayınlar için ise KU30M03 (Ek 4.35) panelleri ekrana gelmektedir. Her iki panelde de kayıt işlemi PF4 tuşuyla yapılmaktadır. PF1 tuşuna basarak KU30H01 (Ek 4.36), KU30H02 (Ek 4.37), KU30H03 (Ek 4.38) ve KU30H04 (Ek 4.39.) yardım panellerine erişilmektedir. Record olarak KU30R01, working storage olarak da KU30W kullanılmıştır.

2-KU03M01 panelinden yapılan seçim 2 ise KU31A programı çalışmaktadır. Akış şeması Ek 5.16. ve 'de, yapısı ise Ek 3.15.'de verilmiştir. Amacı yapılacak sipariş bilgilerinin değişiklik ve silme işlemlerini yapmaktır. KU31M01 (Ek 4.40.) panelinden yapılacak seçimlere göre KU31M02 (Ek 4.41.) paneli görüntülenmektedir. Gerekli bilgilerin girilip Enter tuşuna basılması halinde bulunan bilgiler KU31M03 (Ek 4.42.) paneline getirilmektedir. Ekranlarda ileriye doğru hareket etmek için PF8, geriye doğru hareket etmek için PF7 tuşuna basmak gerekmektedir. PF4 tuşuna basılması halinde yayın kitap ise KU31M04 (Ek 4.43.), süreli yayın ise KU31M05 (Ek 4.44.) paneli görüntülenmekte, gelen bilgiler üzerinde PF5 tuşu ile değişiklik, PF6 tuşu ile silme işlemi yapılmaktadır. KU31M04 panelinde iken PF7 tuşu ile o siparişi veren birime, yayının durumu hakkında birim yazısı gönderilmektedir. Sipariş geldiğinde uygulanacak işlem kodları için Ek 1.9'daki KU13T02 CSP tablosuna bakınız. Program record' KU31R01, working storage'ı ise KU31W'dir. PF1 tuşuna basılarak yardım paneli olan KU31H01'e (Ek 4.45.) ulaşılmaktadır.

3-KU03M01 panelinden 3 seçilmiş ise KU32A programı çalışmaktadır. Akış şeması Ek 5.17.'de, yapısı ise Ek 3.16.'de verilmiştir. Amacı yapılacak sipariş bilgilerinin izlenmesini sağlamaktır. İlk ekran olan KU32M01 (Ek 4.46.) panelinde yapılacak işlemler gösterilmiştir. Bulunan bilgiler KU32M02 (Ek 4.47.) paneline KU32R01 record'u

yardımıyla taşınmaktadır. PF4 tuşu ile siparişin ayrıntılarına erişilmekte, PF5 tuşu ile sipariş bilgilerinin yazdırılması sağlanmaktadır. PF1 tuşuna basılarak KU32H01 (Ek 4.48) yardım paneline erişilmektedir. Kullanılan working storage KU32W'dir. Sipariş listesi yazıcı çıktısı Ek 6.5.'de verilmiştir.

4-KU03M01 panelinden 4 seçilmiş ise KU33A programı çalışmaktadır. Akış şeması Ek 5.18.'de, yapısı ise Ek 3.17.'de verilmiştir. Program, siparişler için muhasebe işlemlerinin takibini sağlamaktadır. Muhasebe işlemleri Muhasebe SQL tablosuna (Ek 2.2.) kaydedilmektedir. Ekranı gelen KU33M01 (Ek 4.49.) panelinde sipariş bilgilerine hangi yollardan erişilebileceği gösterilmiştir. Tercih yapıp Enter tuşuna basılarak KU33M02 (Ek 4.50) paneline gidilmekte, gerekli bilgiler girilip yine Enter tuşuyla sorgulama yapılabilir. Bulunan bilgiler KU33M03 (Ek 4.51) paneline KU33R01 record'u yardımıyla taşınmaktadır. Bu panelde hangi yayına ait muhasebe işlemleri takip edilecekse, kursor o yayının üzerinde iken PF4 tuşuna basılarak muhasebe bilgi giriş ekranı olan KU33M04 (Ek 4.52) paneline geçilmektedir. Bilgiler KU33R02 recordu kullanılarak bu panele getirilmektedir. Enter tuşu ile bilgilerin kaydı yapılmakta, PF9 tuşu ile ekrana gelen bilgilerin yazıcıdan alınması sağlanmaktadır. PF1 tuşuna basılması halinde KU33H01 (Ek 4.53.) yardım panelini çağırır. Working Storage olarak KU33W kullanılmıştır. Ek 1.8.'deki KU13T01 CSP tablosu para birim kodlarını kontrol etmek amacıyla kullanılmıştır. Muhasebe işlemlerine ait yazıcı çıktısı Ek 6.6.'da yer almaktadır.

5-KU03M01 panelinden 5 seçilmiş ise KU34A programı çalışmaktadır. Akış şeması Ek 5.19.'da, yapısı ise Ek 3.18.'de verilmiştir. Sağlama servisinde faturaların takibi amacıyla kullanılmaktadır. Ekranı gelen KU34M01 (Ek 4.54.) paneline gerekli bilgiler girilip Enter tuşu ile işlemler başlatılmaktadır. KU34M02 (Ek 4.55.) paneline gerekli bilgilerin girilip sorgulanmaları yine Enter tuşu ile yapılmaktadır. Bulunan bilgiler KU34R01 record'u yardımıyla KU34M03 (Ek 4.56.) paneline taşınmaktadır. PF9 tuşu ile ekran bilgileri yazıcıdan alınabilmektedir. Yardım ekranı KU34H01 (Ek 4.57) istenirse PF1 tuşuna basılarak görüntülenmektedir. Working storage olarak KU34W kullanılmıştır. Yazıcı çıktısı örneği Ek 6.4.'de yer almaktadır.

6-KU03M01 panelinden yapılan seçim 6 ise KU35A programı çalışmaktadır. Akış şeması Ek 5.20.'da, yapısı ise Ek 3.19.'da verilmiştir. Sağlama servisinde faturaların parasal bazda dağılımını takip etmek amacıyla kullanılmaktadır. Ekranı gelen KU35M01 (Ek 4.58.) panelinde gerekli tercih yapıp -istenirse belli bir yıl da girilerek- Enter tuşuna basılırsa gerekli bilgiler KU35M02 (Ek 4.59.) panelinde sunulmaktadır. PF9 tuşu ile bu bilgiler yazıcıdan alınabilmektedir. Süreli yayınlar için kullanılan record KU35R01, kitap siparişleri için kullanılan record ise KU35R02'dir. Working storage olarak KU35W kullanılmıştır. Ek 6.3.'de örnek yazıcı çıktısı yer almaktadır.

7-KU03M01 panelinden 7 seçilmiş ise KU36A programı çalışmaktadır. Akış şeması Ek 5.21.'de, yapısı ise Ek 3.20.'de verilmiştir. Faturaların firma bazında dağılımını takip etmek için kullanılmaktadır. KU36M01 (Ek 4.60.) paneli gerekli bilgilerle görüntülenmektedir. KU36R01 record olarak, KU36W working storage olarak kullanılmıştır. KU13T03 (Ek 1.10) CSP tablosu firma adı bilgilerini ihtiva etmektedir.

8-KU03M01 panelinden 8 seçilmiş ise **KU37A** programı çalışmaktadır. Akış şeması Ek 5.22.'de, yapısı ise Ek 3.21.'de verilmiştir. Yayınların firma bazında veya alfabetik olarak listelenmesi işlemlerini yapmakta olan programdır. Ekrana **KU37M01** (Ek 4.61.) paneli gelmekte, seçilen tercihle alakalı listeler yazıcıdan alınabilmektedir. **KU37R01** record, **KU37W** working storage'dır. Ek 6.1.'de alfabetik, Ek 6.2.'de firma adına göre çıktı örnekleri bulunmaktadır.

9-KU03M03 panelinin son tercihi olan 9 seçilirse ana menü olan **KU00A**'ya dönmektedir.

2.10.5. KU04A Uygulama Programı

KU04A programı üyelik ve ödünç verme servisinin yapacağı işlemleri çağırılmaktadır. **KU04M01** panelinden yapılacak tercihlere göre işlem yapılmaktadır.

1-KU04M01 panelinden yapılan seçim 1 ise **KU40A** programı çalışmaktadır. Akış şeması Ek 5.23.'de, yapısı ise Ek 3.22.'de verilmiştir. Amacı kütüphaneden faydalanacak okuyucuların kayıtlarının yapılmasını sağlamaktır. Bu bilgiler Üyeler SQL tablosuna kaydedilmektedir. Üyeler SQL tablosuna ait yapı Ek 2.6.'da yer almaktadır. Ekrana gelen **KU40M01** (Ek 4.62.) panelinden yapılacak işlemler şunlardır. Gerekli bilgiler girilip **PF4** tuşu ile kayıt yapılmakta, ad-soyad veya sicil no/öğrenci no girilip Enter tuşuna basılarak daha önceden girilmiş bir üye kaydına erişilmektedir. Gelen bilgiler üzerinde **PF5** tuşu ile değişiklik, **PF6** tuşu ile silme işlemleri yapılabilmektedir. Üye kaydının silinmesi ancak Ödünç SQL tablosunda o üyeye ait bilgi yoksa veya başka bir deyişle üzerindeki tüm yayınları iade etmişse **KU41R02** yardımıyla silinmektedir. Teslim etmediği yayınlar ise üye kaydı silinmemektedir. Yardım bilgilerine ulaşmak için **PF1** tuşuna basılmakta, ekrana **KU41H01** (Ek 63.) ve **KU41H02** (Ek 4.64.) panelleri gelmektedir. Record olarak **KU40R01**, working storage olarak **KU40W** kullanılmıştır. **KU11T06** (Ek 1.6.) CSP tablosu üye ünvan kodlarını göstermektedir.

2-KU04M01 panelinden 2 nolu tercih seçilmiş ise **KU41A** programı çalışmaktadır. Bu program hem okuyuculara yayınların ödünç verilmesi hem de verilen yayınların iadesi amacıyla kullanılmaktadır. Programın akış şeması Ek 5.24.'de, yapısı ise Ek 3.23.'de verilmiştir. Ekrana gelen **KU41M01** (Ek 4.65.) panelinde üzerinde işlem yapılacak kişiye Ad veya Soyadı bilgileri girilip Enter tuşuna basmak suretiyle erişilmektedir. Akabinde gelen **KU41M02** (Ek 4.66.) panelinde yayın okuyucuya teslim edilecekse "O", yayını teslim edecekse "G" harflerini kodlanmalı, kursor okuyucu ismi üzerinde iken **PF4**'e basılması halinde birinci durumda **KU42A**, ikinci durumda **KU43A** programları devreye girmektedir. **KU41A** programında **KU41R01** record, **KU41W** working storage olarak kullanılmıştır.

2.a-KU41M02 panelinde "O" kodlanmış ise, **KU42A** programı yayınların ödünç verilmesi amacıyla devreye girmektedir. Akış şeması 5.25.'de, program yapısı Ek 3.24.'de verilmiştir. **KU42M01** (Ek 4.67.) paneline gerekli bilgiler girilip **PF4** tuşu ile kayıt işlemleri yapılmaktadır. **PF1** tuşu ile **KU42H01** (Ek 4.68.) yardım paneline erişilmektedir. Working storage **KU42W**, record **KU42R01** ve **KU42R02** dir. Ödünç verme işlemleri için geliştirilen form örneği Ek 6.20. ve Ek 6.21.'de verilmiştir.

2.b-"G" olarak kodlanan KU41M02 paneli ise, KU43A programını ödünç verilen yayınların geri alınması işlemlerini yapmak üzere çağıracaktır. Program yapısı Ek 3.25.'de verilirken, akış şeması Ek 5.26.'da verilmiştir. KU43M01 (Ek 4.69.) sadece iade tarihi girilmesi yeterli olmaktadır. Yardım bilgilerine erişmek için PF1 tuşuna basılırsa KU43H01 (Ek 4.70.) paneli görüntülenmektedir. Record olarak KU43R01, KU43R02 ve KU43R03 kullanılmıştır. Working Storage adı ise KU43W'dir. PF5 tuşu ile ödünç verilen yayınlar yazıcıdan alınabilmektedir. Örnek yazıcı çıktısı Ek 6.7.'de verilmiştir.

3-KU04M01 panelinin 3. seçeneği, süresini geçirmiş olan (Kitaplarda 30 gün, süreli yayınlarda 15 gün) şahısları tesbit eden KU44A programını çalıştırmaktadır. Yapısı Ek 3.26.'da, akış şeması Ek 5.27.'de verilmiştir. Program çalıştığında KU44M01 (Ek 4.71.) paneline gelen bilgilerde kursor Kart No alanı üzerinde iken PF4 tuşuna basılırsa yayını alan şahsın bilgileri ve geçen süre ekrana gelmektedir. PF5 tuşu ile de ekran bilgileri yazıcıdan alınmakta, PF1 tuşu ile yardım ekranı olan KU44H01 (Ek 4.72.) görüntülenmektedir. Working Storage olarak KU44W, record olarak KU44R01 ve KU44R02 kullanılmıştır.

4-KU04M01 ekranında iken 4 nolu tercih seçilmişse KU45A programı ödünç verilen herhangi bir yayını aramak amacıyla çağırılmaktadır. Akış şeması Ek 5.28.'de, program yapısı Ek 3.27.'de verilmiştir. Yayın adı, yıl, cilt veya sayı bilgileri KU45M01 (Ek 4.73.) panelinden girilerek Enter tuşuna basılmak suretiyle arama işlemleri başlatılmaktadır. KU45M02 (Ek 4.74.) paneline taşınan bilgilerde, kursor herhangi bir yayın üzerinde iken PF9 tuşu ile o yayının kimde olduğu görülebilmektedir. Yardım panelinin adı KU45H01 (Ek 4.75.) dir. KU45R01 record olarak, KU45W working storage olarak kullanılmıştır.

5-KU46A programı KU04M01 panelinden 5. seçenek tercih edilmişse çalışmaktadır. Amacı ödünç verilen yayınların okuyucu grupları arasındaki dağılımını tesbit etmektir. KU46A programının yapısı Ek 3.28.'de, akış şeması Ek 5.29.'da verilmiştir. KU46M01 (Ek 4.76.) paneli kullanıcı gruplarını ve alış sayılarını göstermektedir. Record KU46R01, working storage KU46W'dir.

6-KU04M01 panelinin 6. seçeneği KU47A programını çalıştırmaktadır. Program herhangi bir yıl veya aya ait yapılan işlem sayısını, alınıp iade edilen ve iade edilmeyen yayın sayılarını göstermektedir. Akış şeması Ek 5.30.'da, yapısı Ek 3.29.'da verilmiştir. Ekran paneli KU47M01 (Ek 4.77.) de gösterilmiştir. Record olarak KU47R01, working storage olarak KU47W kullanılmıştır.

7-KU04M01 panelinin 7. seçeneği ana menüye dönüşü temin etmektedir.

2.10.6. KU05A Uygulama Programı

KU05A programı dökümantasyon servisinin yapacağı işlemler için hazırlanmıştır. KU05M01 panelinden yapılacak tercihlere göre işlem yapmaktadır.

1-KU05M01 panelinin ilk seçeneği KU51A programını çalıştırmaktadır. Amacı 1/25.000 ölçekli harita bilgilerinin Ybesbin SQL tablosuna (Ek 2.7.) kaydedilmesini sağlamaktır. Bu Akış şeması Ek 5.31.'de, program yapısı Ek 3.30.'da verilmiştir. Record olarak

gerekli bilgiler girilip Enter tuşu ile tarama işlemleri başlatılmaktadır. Working storage olarak TA21W ve TA21W1 kullanılmıştır. Bulunan bilgiler TA21R01 yardımıyla TA21M02 (Ek 4.89.) paneline aktarılmaktadır. Bu bilgiler PF5 tuşu ile istenildiği takdirde yazıcıdan alınabilmektedir. Kursor herhangi bir yayın üzerinde iken PF4 tuşuna basılması halinde yayın hakkında daha ayrıntılı bilgiler katalog kartı düzeninde ekrana getirilmekte, yayın şu anda nerede olduğu, ödünç olarak verilip verilmediği gibi bilgiler de ekran üzerinde gösterilmektedir. PF1 tuşuna basılması halinde TA21H01 (Ek 4.90.), TA21H02 (Ek 4.91.), TA21H03 (Ek 4.92.), TA21H04 (Ek 4.93.) yardım panellerine erişilebilmektedir.

2-TA10M01 panelinden 1 nolu tercih seçilmişse TA30A programı çalışmakta ve TA30M01 (Ek 4.85.) paneli görüntülenmektedir. Bu bölüm süreli yayın taramalarına ait işlemlerini yapmaktadır. Yapılan seçime göre hangi program bölümünün devreye gireceği TA31T02 CSP tablosundan kontrol edilmektedir. Kullanılan working storage TA30Wdir. Programın yapısı Ek 3.36.'da, akış şeması Ek 5.38.'de verilmiştir.

TA30A, program akışını TA31A programına bırakmaktadır. TA31A'nın akış şeması Ek 5.39.'da, program yapısı Ek 3.40.'da verilmiştir. TA31M01 (Ek 4.94.) paneline, gerekli bilgiler girilip Enter tuşu ile tarama işlemleri başlatılmaktadır. TA31W ve TA31W1 working storage olarak kullanılmıştır. Bulunan bilgiler TA31R01 yardımıyla TA31M02 (Ek 4.95.) paneline aktarılmaktadır. Bu bilgiler PF5 tuşu ile istenildiği takdirde yazıcıdan alınabilmektedir. Kursor herhangi bir yayın üzerinde iken PF4 tuşuna basılması halinde yayın hakkında daha ayrıntılı bilgiler Süreli Yayın Katalog Kartı düzeninde ekrana getirilmektedir. Ekranlarda PF1 tuşuna basılması halinde TA31H01 (Ek 4.96.), TA31H02 (Ek 4.97.), TA31H03 (Ek 4.98.), TA31H04 (Ek 4.99.) yardım panellerine erişilebilmektedir.

3-TA10M01 panelinden 3 nolu tercih seçilmişse TA40A programı çalışmakta ve TA40M01 (Ek 4.86.) paneli görüntülenmektedir. Bu bölüm tez-makale taramalarını yamak üzere hazırlamıştır. Yapılan seçime göre hangi program bölümünün devreye gireceği TA41T01 CSP tablosundan kontrol edilmektedir. Working storage TA40Wdir. Programın yapısı Ek 3.37.'da, akış şeması Ek 5.40.'de verilmiştir.

TA40A, program akışını TA41A programına devretmektedir. TA41A'nın akış şeması Ek 5.41.'de, program yapısı Ek 3.41.'da verilmiştir. TA41M01 (Ek 4.100.) paneline, gerekli bilgiler girilip Enter tuşu ile tarama işlemleri başlatılmaktadır. TA41W ve TA41W1 working storage olarak kullanılmıştır. Bulunan bilgiler TA41R01 yardımıyla TA41M02 (Ek 4.101.) paneline aktarılmaktadır. Bu bilgiler PF5 tuşu ile istenildiği takdirde yazıcıdan alınabilmektedir. Kursor herhangi bir yayın üzerinde iken PF4 tuşuna basılması halinde yayın hakkında daha ayrıntılı bilgiler tez ve makalelere ait katalog kartı düzeninde ekrana getirilmektedir. Ekranlarda PF1 tuşuna basılarak TA41H01 (Ek 4.102.), TA41H02 (Ek 4.103.), TA41H03 (Ek 4.104.), TA41H04 (Ek 4.105.) yardım panellerine erişilebilmektedir.

4-TA10M01 panelinden 4 nolu tercih seçilmişse TA50A programı çalışmakta ve TA50M01 (Ek 4.87.) paneli görüntülenmektedir. Harita bilgileri taramalarını yapan bir

programdır. Yapılan seçime göre hangi program bölümünün devreye gireceği TA51T01 CSP tablosundan kontrol edilmektedir. Working storage adı TA50Wdir. Programın yapısı Ek 3.38.'de, akış şeması Ek 5.42.'de verilmiştir.

TA50A, program akışını TA51A programına devretmektedir. TA51A'nın akış şeması Ek 5.43.'de, program yapısı Ek 3.42.'de verilmiştir. TA51M01 (Ek 4.106.) paneline, gerekli bilgiler girilip Enter tuşu ile tarama işlemleri başlatılmaktadır. Working Storage olarak TA51W kullanılmıştır. Bulunan bilgiler TA51R01 yardımıyla TA51M02 (Ek 4.107.) paneline aktarılmaktadır. Bu bilgiler PF5 tuşu ile istenildiği takdirde yazıcıdan alınabilmektedir. PF1 tuşuna basılarak TA51H01 (Ek 4.108.) yardım paneline erişilmektedir.

5-TA10M01 panelinin 5 nolu tercihi ile program bitirilmekte CMS ortamına dönülmektedir.



EKLER

Ek 1. Kullanılan CSP Tabloları	(s.109)
Ek 2. Kullanılan SQL Tabloları	(s.112)
Ek 3. Program Yapıları	(s.116)
Ek 4. Program Ekranları	(s.135)
Ek 5. Program Akış Şemaları	(s.189)
Ek 6. Program Yazıcı Çıktıları	(s.236)

Ek 1. Programda Kullanılan CSP Tabloları

TÜRÜ	İZAH
KT	Kitap
PT	Patent ve Copyright
TR	Teknik Resim
RS	Resim
EY	El Yazması
NE	Nadir Eser
MU	Müracaat Eseri
TZ	Tez
MK	Makale

Ek 1.1. Yayın Türü Kodları (KU11T01 Tablosu)

DİL KODU	DİL ADI
TUR	Türkçe
İNG	İngilizce
ALM	Almanca
OSM	Osmanlıca
FRA	Fransızca
ARP	Arapça
RUS	Rusça

Ek 1.2. Dil Türü Kodları (KU11T02 Tablosu)

ORTAM	ORTAM ADI
NR	Normal Kağıt
FL	Film
MF	Mikro Fiş
SL	Slayt
CD	Compact Disc
VI	Video Film

Ek 1.3. Kayıt Ortamı Kodları (KU11T03 Tablosu)

ÇIKIŞ KODU	ACIKLAMASI	ÇIKIŞ KODU	ACIKLAMASI
1	Yılda 1 defa	11	Yılda 11 defa
2	Yılda 2 defa	12	Yılda 12 defa
3	Yılda 3 defa	24	Yılda 24 defa
4	Yılda 4 defa	36	Yılda 36 defa
5	Yılda 5 defa	48	Yılda 48 defa
6	Yılda 6 defa	52	Yılda 52 defa
7	Yılda 7 defa	G.	Günlük
8	Yılda 8 defa	D.	Düzensiz
9	Yılda 9 defa		
10	Yılda 10 defa		

Ek 1.4. Süreli Yayınlar Çıkış Periyotları (KU11T04 Tablosu)

POZİSYON	ANLAMI
0	Kütüphanede
1	Okuyucuda
2	Depoda
3	Belli Değil

Ek 1.5. Yayın Durum Kodları (KU11T05 Tablosu)

KODU	UNVAN
1	Profesör
2	Yrd. Doç.
3	Doçent
4	Doktor
5	Doktora Öğrencisi
6	Araştırma Görevlisi
7	Lisans Öğrencisi
8	Yüksek Lisans Öğrencisi
9	Memur
U	Uzman
O	Okutman

Ek 1.6. Üye Ünvan Kodları (KU11T06 Tablosu)

TÜR	TÜR İZAH
1	Yüksek Lisans
2	Doktora Tezi
3	Tıpta Uzmanlık
4	Doçentlik Tezi
5	Bitirme Tezi

Ek 1.7. Tez-Makale Tür Kodları (KU11T07 Tablosu)

PARAKODU	PARABİRİMİ
1	USD (Amerikan Doları)
2	DM (Alman Markı)
3	UKS (İngiliz Sterlini)
4	TL (Türk Lirası)
5	HF (Hollanda Florini)
6	JY (Japon Yeni)

Ek 1.8. Para Birim Kodları (KU13T01 Tablosu)

GELİŞİŞİSİEMKODU	ACIKLAMALARI
1	Kitap veya Yazar adı yanlış yazılmıştır.
2	Adres yetersizdir.
3	Mevcudu yoktur. Sipariş iptal edilmiştir.
4	Basılmaktadır.
5	Yayını kesilmiştir.
6	Kitap gelmiştir.
7	Periyodik gelmektedir.
8	Yayınevi aboneyi kabul etmiyor
9	İstenilen bilgi 5 gün içinde verilmesse sipariş iptal edilir.

Ek 1.9. Sipariş Geldiğinde Uygulanacak İşlem Kodları (KU13T02 Tablosu)

F KODU	F ADI
1	Swets
2	Blackwell's
3	Ebsco
4	F.Delbanco
5	Infotel

Ek 1.10. Firma Kodları (KU13T03 Tablosu)

TABLO ADI	S E C İ M							
	1	2	3	4	5	6	7	8
APPT01	KU01A	KU02A	KU03A	KU04A	KU05A	-	-	-
APPT02	KU10A	KU11A	KU12A	-	-	-	-	-
APPT03	KU20A	KU21A	KU22A	-	-	-	-	-
APPT04	KU30A	KU31A	KU32A	KU33A	KU34A	KU35A	KU36A	KU37A
APPT05	KU40A	KU41A	KU42A	KU45A	KU46A	KU47A	-	-
APPT06	KU51A	KU52A	KU53A	KU54A	-	-	-	-
TA10T01	TA20A	TA30A	TA40A	TA50A	-	-	-	-

Ek 1.11 CSP Tablosu / Uygulama Adı İlişkisi

Ek 2. Programda Kullanılan SQL Kütüklerinin Yapısı

KOLON ADI	TİPİ
SIRA_NO *	INTEGER
BELGE_TURU	CHAR (2)
KONU_NO	CHAR (20)
TEZ_KOD	CHAR
ISBN_NO	CHAR (18)
YAZAR_KOD	CHAR (5)
AYIRICI_KOD	CHAR (4)
YAZAR_ADI	CHAR (15)
YAZAR_SOYADI	CHAR (25)
EMEGI_GECENLER	CHAR (140)
KITAP_ADI	CHAR (140)
KONU_ADI	CHAR (140)
ADEDİ	INTEGER
YAYIN_YERİ	CHAR (40)
YAYINLAYAN	CHAR (50)
YAYIN_YILI	INTEGER
SERİ_ADI	CHAR (70)
BİBLİYOGRAFYA	CHAR (25)
NOT_ALANI	CHAR (50)
INDEX_ALANI	CHAR (20)
SAYFACILT_S	CHAR (25)
FİZİKİ_OZ	CHAR (50)
BASKI_KAYDI	CHAR (20)
KAYIT_ORTAMI	CHAR (2)
DİLİ	CHAR (3)
DEMİRBAS_NO	CHAR (4)
İZLEME	CHAR (140)
ODUNC_VERME	CHAR
KONUMU	CHAR
DBASNO	CHAR (140)

Ek 2.1. Kitaplar SQL tablosunun yapısı (* işaretli alanlar Primary Key sahalarıdır.)

KOLON ADI	TİPİ
SIRA_NO *	INTEGER
ISBN_ISSN	CHAR (18)
ÇIK_YILI	INTEGER
FAT_TURU	CHAR
KAYIT_BİLGİSİ	CHAR
FATURA_N	CHAR (10)
FATURA_F	DECIMAL (10,2)
FATURA_T	DATE
VOLUME	CHAR (10)
PARA_BİRİMİ	CHAR (3)
FİRMA_KODU	CHAR

Ek 2.2. Muhasebe SQL tablosunun yapısı

KOLON ADI	TİPİ
ISMAR NOSU *	INTEGER
ISMAR TARİHİ	DATE
KAYIT BİLGİSİ	CHAR
PARA BİRİMİ	CHAR (3)
DİLİ	CHAR (3)
ÇIKIS ARALIGI	CHAR (5)
KONU NO	CHAR (20)
ISBN ISSN	CHAR (18)
FAKYUKENST	CHAR (60)
BOLUM	CHAR (60)
IST TAR	DATE
EVSAI	CHAR (30)
YAZAR	CHAR (60)
KITAPAD	CHAR (60)
C SERİ	CHAR (30)
VOL SAYI	CHAR (30)
FİZİKİ ÖZ	CHAR (50)
EDİSYON	CHAR (20)
YAYINLAYANI	CHAR (50)
YAYINYERİ	CHAR (50)
YAYIN TARİH	INTEGER
BİRFİATİ	DECIMAL (10,3)
GELİŞ TARİH	INTEGER
SATAN	CHAR (60)
TOPMALİYETİ	DECIMAL (10,3)
GEL YAP İS	CHAR (2)
FATURA NO	CHAR (18)
DEMİRBASI	CHAR (18)
ESKİ AD11	CHAR (60)
ESKİ AD12	CHAR (60)
ESKİ AD13	CHAR (60)
ESKİ ISSN1	CHAR (18)
ESKİ ISSN2	CHAR (18)
ESKİ ISSN3	CHAR (18)
ABONE DUR	CHAR

Ek 2.3. Kayıt SQL tablosunun yapısı

KOLON ADI	TİPİ
SIRA NO *	INTEGER
ISSN_NO	CHAR (18)
ÇIK YILI	INTEGER
VOLUM	CHAR (4)
AY BİLGİSİ	CHAR (2)
SY BİLGİSİ	CHAR (4)
INDEXİ	CHAR (6)
SAYILARI	CHAR (62)

Ek 2.4. Eksik_sayı SQL tablosunun yapısı

KOLON ADI	TİPİ
SIRA NO *	INTEGER
KART NO *	INTEGER
SICIL OGNO	CHAR (9)
DERKIT ADI	CHAR (3)
YAYIN YIL	INTEGER
CILT	INTEGER
SAYI	CHAR (25)
ALIS TAR	DATE
IADE TAR	DATE
UNVANI	CHAR

Ek 2.5. Odunc SQL tablosunun yapısı

KOLON ADI	TİPİ
KART NO *	INTEGER
KAYIT TAR	DATE
SICIL OGNO	CHAR (9)
ADI	CHAR (15)
SOYADI	CHAR (20)
BOLUMU	CHAR (40)
TELEFON	CHAR (15)
UNVANI	CHAR
ALIS SAYISI	INTEGER
UYELIK DURUMU	CHAR

Ek 2.6. Uyeler SQL tablosunun yapısı

KOLON ADI	TİPİ
SIRA NO *	INTEGER
HARITA ADI	CHAR (15)
ALAN1	CHAR (8)
ALAN2	CHAR (8)
ALAN3	CHAR (8)
ALAN4	CHAR (8)
TOPLAM	INTEGER

Ek 2.7. Ybesbin SQL tablosunun yapısı

KOLON ADI	TİPİ
SIRA NO *	INTEGER
ILIN ADI1	CHAR (15)
ILIN ADI2	CHAR (15)
PAFTA NO1	CHAR (3)
PAFTA NO2	CHAR (3)
TOPLAM	INTEGER

Ek 2.8. Yuzbin SQL tablosunun yapısı

KOLON ADI	TİPİ
SIRA NO *	INTEGER
ILIN ADI	CHAR (15)
PLAKA NO	CHAR (3)
TOPLAM	INTEGER

Ek 2.9. İkielli SQL tablosunun yapısı

KOLON ADI	TİPİ
SIRA NO *	INTEGER
EBAT	CHAR (6)
OZEL NO	CHAR (15)
NOTU	CHAR (30)
TOPLAM	INTEGER

Ek 2.10. Havafot SQL tablosunun yapısı

SQL KÜTÜĞÜ	GÖREVİ
Kitaplar	Kataloglama (Kitap-Tez-Makale) Servisi Hizmetleri
Eksik sayı	Sürelî Yayın Servis Hizmetleri
Kayıt	Sağlama Servisi Hizmetleri
Muhasebe	Sağlama Servisi Muhasebe Hizmetleri
Uyeler	Ödünç Verme Servisi Üyelik Hizmetleri
Odunc	Ödünç Verme Servisi Yayın Giriş-Çıkış Hizmetleri
Ybesbin	Dökümantasyon Servisi 1/ 25.000 Ölçekli Harita Hizmetleri
Yuzbin	Dökümantasyon Servisi 1/105.000 Ölçekli Harita Hizmetleri
İkielli	Dökümantasyon Servisi 1/250.000 Ölçekli Harita Hizmetleri
Havafot	Dökümantasyon Servisi Hava Fotoğrafları Hizmetleri

Ek 2.11. Uygulamalarda kullanılan SQL kütüklerinin görevleri

Ek 3. Uygulama Programına Ait Yapılar

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU00P01	001	Execute			Kütüphane İşlemleri Seçimi
KU00P02	002	Converse	KU00M01		Kütüphane İşlemleri
KU00P03	002	Converse	KU00M02		Yetkisiz Kullanıcı Girişi

Ek 3.1. Ana Menü Uygulama Yapısı (KU00A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU01P01	001	Execute			Execute KU01A
KU01P02	002	Converse	KU01M01		Kataloglama Seçenekleri

Ek 3.2. Kataloglama Servisi Uygulama Yapısı (KU01A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU02P01	001	Execute			Execute KU02A
KU02P02	002	Converse	KU02M01		Sürelili Yayın Seçenekleri

Ek 3.3. Sürekli Yayın Servisi Uygulama Yapısı (KU02A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU03P01	001	Execute			Execute KU03A
KU03P02	002	Converse	KU03M01		Sağlama Servisi Seçenekleri

Ek 3.4. Sağlama Servisi Uygulama Yapısı (KU03A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU04P01	001	Execute			Execute KU04A
KU04P02	002	Converse	KU04M01		Üyelik/Ödünç Verme İşlem.

Ek 3.5. Ödünç Verme Servisi Uygulama Yapısı (KU04A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU05P01	001	Execute			Execute KU05A
KU05P02	002	Converse	KU05M01		Dökümantasyon İşlemleri

Ek 3.6. Dökümantasyon Servisi Uygulama Yapısı (KU05A)

NAME	LYL	OPTION	OBJECT	ERROR	DESCRIPTION
KU10P01	001	Execute			Execute KU10A
KU10P02	002	Converse	KU10M01		Katalog Bilgi Giriş Ekranı-1
KU10S01	003	Group			Başlangıç Bilgileri
KU10P03	003	Converse	KU10M02		Katalog Bilgi Giriş Ekranı -2
KU10S02	003	Group			Kayıt İşlemine Başlangıç
KU10P04	004	Inquiry	KU10R01	EZERTN	Sıra No'sunu Bul
KU10P05	004	Add	KU10R01	EZERTN	Kayıt Process'i
KU10S01	004	Group			Başlangıç Bilgileri

Ek 3.7. Katalog Servisi Bilgi Giriş Uygulama Yapısı (KU10A)

NAME	LYL	OPTION	OBJECT	ERROR	DESCRIPTION
KU11P01	001	Execute			Execute KU11A
KU11P02	002	Converse	KU11M01		Converse KU11M01
KU11S01	003	Group			Başlangıç
KU11P03	003	Converse	KU11M02		Bilgilerin Girilmesi
KU11P04	004	Setinq	KU11R01	EZERTN	ISBN No'dan Erişim
KU11P05	004	Setinq	KU11R01	EZERTN	Yazar Adı'ndan Erişim
KU11P06	004	Setinq	KU11R01	EZERTN	Yayın Adı'ndan Erişim
KU11P07	004	Scan	KU11R01	EZERTN	Scan KU11R01
KU11S02	005	Group			Record Working Storage'a Taşı
KU11P07	004	Scan	KU11R01	EZERTN	Scan KU11R01
KU11P08	004	Converse	KU11M03		Bilgiler Ekranı
KU11S03	005	Group			Working Storage'ı Ekranı Taşı
KU11S04	005	Group			Ekranı Working Storage'a Taşı
KU11S05	005	Group			Kursurun Tesbiti
KU11S06	006	Group			Kursor Bulundu
KU11P09	007	Inquiry	KU11R01	EZERTN	Sıra No'sunu Bul
KU11S07	007	Group			Kayıt Bilgileri Ekranı
KU11P10	007	Converse	KU11M04		Katalog Bilgileri 1. Ekran
KU11P11	008	Converse	KU11M05		Katalog Bilgileri 2. Ekran
KU11S08	009	Group			Ekran Bilgilerini Record'a Taşı
KU11S09	008	Group			Değişiklik İşlemine Başla
KU11P12	009	Sqlxec	KU11R01	EZERTN	Değişiklik Process'i
KU11S10	008	Group			Silme İşlemine Başla
KU11P13	009	Sqlxec	KU11R01	EZERTN	Silme Process'i

Ek 3.8. Katalog Servisi Bilgi Düzeltme - Silme Uygulama Yapısı (KU11A)

NAME	LYL	OPTION	OBJECT	ERROR	DESCRIPTION
KU12P01	001	Execute			Execute KU12A
KU12P02	002	Converse	KU12M01		Katalog Kartı Bastırma

Ek 3.9. Katalog Servisi Katalog Kartları Uygulama Yapısı (KU12A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU13P01	001	Execute			Execute KU13A
KU13P02	002	Converse	KU13M01		Katalog Seçimi Bilgi Giriş
KU13S01	003	Group			Seçime Bağlı Process'lerin
KU13P03	004	Setinq	KU13R01	EZERTN	Yazar Adından Erişim
KU13P04	004	Setinq	KU13R01	EZERTN	Yayın Adından Erişim
KU13P05	004	Setinq	KU13R01	EZERTN	Konu Adından Erişim
KU13P06	004	Setinq	KU13R01	EZERTN	Dizi (Seri) Adından Erişim
KU13P07	004	Setinq	KU13R01	EZERTN	Yer Numarasından Erişim
KU13P08	004	Scan			Scan KU13R01
KU13P09	004	Converse	KU13M02		Bilgiler Ekrana
KU13S02	005	Group			Working Storage'ı Ekrana Taşı
KU13S03	005	Group			Ekranı Working Storage'a Taşı
KU13S04	005	Group			Bilgilerin Alınması
KU13P10	006	Inquiry	KU13R01	EZERTN	Sıra No'dan Inquiry
KU13S05	006	Group			Kayıt Working Storage'a
KU13S06	007	Group			Son Noktaların Tesbiti
KU13S07	006	Group			Bilgilerin Aktarımı
KU13S08	006	Group			Yapılara Aktarım
KU13P11	007	Display	KU13M04		Detay Bilgi Ekranı
KU13P13	007	Display	KU13M05		Demirbaş Numarasını Yazdır
KU13P15	007	Display	KU13M07		Belge T., Yazar K., Konu No
KU13P12	007	Display	KU13M03		Detaylar
KU13P14	007	Display	KU13M06		Adedi Yazdır

Ek 3.10. Katalog Servisi Katalog Kartları Bastırma Uygulama Yapısı (KU13A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU20P01	001	Execute			Execute KU20A
KU20P02	002	Converse	KU20M01		Sürelî Yayın Seçenekleri
KU20P03	003	Converse	KU20M02		Converse KU20M02
KU20P04	004	Setinq	KU20R01	EZERTN	ISSN No Erişim
KU20P05	004	Setinq	KU20R01	EZERTN	Yayın Adından Erişim
KU20P06	004	Setinq	KU20R01	EZERTN	Firma Adından Erişim
KU20P07	004	Scan	KU20R01	EZERTN	Scan KU20R01
KU20S01	004	Group			Kayıt Working Storage'a
KU20P07	005	Scan	KU20R01	EZERTN	Scan KU20R01
KU20P08	004	Converse	KU20M03		Sürelî Yayınlar Ekrana
KU20S02	005	Group			Working Storage'ı Ekrana
KU20S03	005	Group			Ekranı Working Storage'a
KU20S04	005	Group			Başlangıç Bilgileri
KU20S05	005	Group			Kursorun Tesbiti
KU20S06	006	Group			Kursor Bulundu
KU20P09	007	Inquiry	KU20R01	EZERTN	ISSN - Sıra No'suna Göre
KU20P10	005	Converse	KU20M04		Sürelî Yayın Bilgileri
KU20S07	005	Group			Bilgileri Taşı
KU20P11	006	Inquiry	KU20R02	EZERTN	Maksimum Sıra No
KU20P12	006	Setinq	KU20R02	EZERTN	Setinq KU20R02
KU20P13	006	Scan	KU20R02	EZERTN	Scan KU20R02
KU20P14	005	Converse	KU20M05		Eksik Sayılar
KU20S08	006	Group			Working Storage'ı Ekrana
KU20S09	006	Group			Ekranı Working Storage'a
KU20S15	006	Group			Kursorun Tesbiti
KU20S16	007	Group			Kursor Bulundu
KU20P15	006	Converse	KU20M06		Eksik Sayılar
KU20S13	007	Group			Working Storage'ı Ekrana
KU20S14	007	Group			Ekranı Working Storage'a
KU20S10	006	Group			İşlemlere İçin Atamalar
KU20P16	007	Add	KU20R02	EZERTN	Kayıt Process'i
KU20P17	007	Sqlxec	KU20R02	EZERTN	Değişiklik Process'i
KU20P18	007	Sqlxec	KU20R02	EZERTN	Silme Process'i
KU20S17	006	Group			Çıkış Aralığı G.-D.-52-24-36
KU20P20	007	Display	KU20M08		Başlık Map
KU20P21	007	Display	KU20M09		Detay Map
KU20P19	005	Converse	KU20M07		Eksik Sayılar
KU20S11	006	Group			Working Storage'ı Ekrana
KU20S12	006	Group			Ekranı Working Storage'a
KU20S10	006	Group			İşlemlere İçin Atamalar
KU20P16	007	Add	KU20R02	EZERTN	Kayıt Process'i
KU20P17	007	Sqlxec	KU20R02	EZERTN	Değişiklik Process'i
KU20P18	007	Sqlxec	KU20R02	EZERTN	Silme Process'i
KU20S18	006	Group			Çıkış Periyodu Aylık
KU20P22	007	Display	KU20M10		Başlık Map
KU20P23	007	Display	KU20M11		Detay Map

Ek 3.11. Sürelî Yayın Servisi Eksik Sayıları Takip Uygulama Yapısı (KU20A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU21P00	001	Execute			Execute KU21A
KU21P01	002	Converse	KU21M01		Giriş Ekranı
KU21P02	003	Setinq	KU21R01	EZERTN	Kitap Adına Göre Sırala
KU21P021	003	Setinq	KU21R01	EZERTN	Girilen Harfe Göre Sırala
KU21P03	003	Scan	KU21R01	EZERTN	Scan KU21R01
KU21P04	003	Display	KU21M02		İlk Ekranı Yaz
KU21S01	003	Group			Eksik Sayı Tablosuna Eriş
KU21P06	004	Setinq	KU21R02	EZERTN	Yayın Yılına Tara
KU21P07	004	Scan	KU21R02	EZERTN	Scan KU21R02
KU21S02	004	Group			Eksik Yıllar Formatı
KU21S03	004	Group			Sayılar Tamam ise
KU21P08	004	Display	KU21M04		2. Ekranı Yazdır
KU21S04	003	Group			Yayının Eski Adı Varsa
KU21P09	004	Display	KU21M05		3. Ekranı Yazdır
KU21P10	004	Display	KU21M06		4. Ekranı Yazdır
KU21P11	004	Display	KU21M07		Sayfa No'sunu Bastır

Ek 3.12. Süreli Yayın Servisi Katalog Listesi Uygulama Yapısı (KU21A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU22P01	001	Execute			Execute KU22A
KU22P02	002	Converse	KU22M01		Tarama Seçeneği Bilgi giriş
KU22S01	003	Group			Setinq & Scan
KU22P03	003	Setinq	KU22R01	EZERTN	Süreli Yayın Adı
KU22P04	003	Scan	KU22R01	EZERTN	Scan KU22R01
KU22S02	003	Group			Yazdırma İşlemi
KU22P05	003	Inquiry	KU22R01	EZERTN	Inquiry Sipariş No
KU22S03	004	Group			Kayıt Working Storage'a
KU22S04	004	Group			Son Noktaların Tesbiti
KU22S05	004	Group			Küçük Harfe Çevir
KU22S06	004	Group			Yapılara Aktar
KU22P06	004	Display	KU22M02		Detay Satırlar

Ek 3.13. Süreli Yayın Servisi Katalog Kartı Uygulama Yapısı (KU22A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU30P01	001	Execute			Execute KU30A
KU30P02	002	Converse	KU30M01		Sipariş Seçimi
KU30S01	003	Group			Sipariş Bilgileri Ekranı
KU30S02	004	Group			Başlangıç Bilgileri
KU30P03	005	Inquiry	KU30R01	EZERTN	Sipariş No'sunun Tesbiti
KU30P04	005	Inquiry	KU30R01	EZERTN	Son Kaydedilen ISBN-ISSN
KU30P05	004	Converse	KU30M02		Kitap Siparişleri
KU30P06	004	Converse	KU30M03		Dergi Siparişleri
KU30S03	004	Group			Kayıt İşlemi İçin Atamalar
KU30P07	005	Add	KU30R01	EZERTN	Siparişlerin Kaydedilmesi
KU30S02	005	Group			Başlangıç Bilgileri
KU30P03	006	Inquiry	KU30R01	EZERTN	Sipariş No'sunun Tesbiti
KU30P04	006	Inquiry	KU30R01	EZERTN	Son Kaydedilen ISBN-ISSN

Ek 3.14. Sağlama Servisi Sipariş Bilgi Girişi Uygulama Yapısı (KU30A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU31P01	001	Execute			Execute KU31A
KU31P02	002	Converse	KU31M01		Arama Seçenekleri
KU31P03	003	Converse	KU31M02		Bilginin Girilmesi
KU31P04	004	Setinq	KU31R01	EZERTN	ISBN No Erişim
KU31P05	004	Setinq	KU31R01	EZERTN	Yazar Adından Erişim
KU31P06	004	Setinq	KU31R01	EZERTN	Kitap Adından Erişim
KU31P07	004	Setinq	KU31R01	EZERTN	Fakülte Adından Erişim
KU31P08	004	Setinq	KU31R01	EZERTN	Bölüm Adından Erişim
KU31P09	004	Scan	KU31R01	EZERTN	Scan KU31R01
KU31S01	004	Group			Kayıt Working Storage'a
KU31P09	005	Scan	KU31R01	EZERTN	Scan KU31R01
KU31P10	004	Converse	KU31M03		Sipariş Bilgileri Ekranı
KU31S02	005	Group			Working Storage'ı Ekranı Taşı
KU31S03	005	Group			Ekranı Working Storage'a Taşı
KU31S04	005	Group			Kursorun Tesbiti
KU31S05	006	Group			Kursor Bulundu
KU31S06	007	Group			Kayıt Bilgileri Ekranı
KU31P11	008	Inquiry	KU31R01	EZERTN	Sipariş No'dan Erişim
KU31P12	007	Converse	KU31M04		Kitap Siparişleri
KU31P13	007	Converse	KU31M05		Dergi Siparişleri
KU31S07	007	Group			Değişiklik İçin Atamalar
KU31P14	008	Sqlexec	KU31R01	EZERTN	Değişiklik Process'i
KU31S08	007	Group			Silmek İçin Atamalar
KU31P15	008	Sqlexec	KU31R01	EZERTN	Silme Process'i
KU31S09	007	Group			İlgili Birime Yazı Gönderilmesi
KU31P16	008	Display	KU31M06		Bilginin Yazıcıdan Alınması

Ek 3.15. Sağlama Servisi Sipariş Değişiklik-Silme Uygulama Yapısı (KU31A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU32P01	001	Execute			Execute KU32A
KU32P02	002	Converse	KU32M01		İzleme Seçenekleri
KU32P03	003	Setinq	KU32R01	EZERTN	Siparişi Gelenler
KU32P04	003	Setinq	KU32R01	EZERTN	Siparişi Gelmeyenler
KU32P05	003	Setinq	KU32R01	EZERTN	Tüm Siparişler
KU32P06	003	Scan	KU32R01	EZERTN	Scan KU32R01
KU32P07	003	Converse	KU32M02		Siparişler Ekrana
KU32S01	004	Group			Kayıt Working Storage'a
KU32P06	005	Scan	KU32R01	EZERTN	Scan KU32R01
KU32S02	004	Group			Working Storage'ı Ekrana Taşı
KU32S03	004	Group			Ekranı Working Storage'a Taşı
KU32S04	004	Group			Kursorun Tesbiti
KU32S05	005	Group			Kursor Bulundu
KU32S06	006	Group			Kayıt Bilgileri Ekrana
KU32P08	007	Inquiry	KU32R01	EZERTN	Sipariş No Inquiry
KU32P09	007	Converse	KU32M03		Kitap Siparişleri
KU32P10	007	Converse	KU32M04		Dergi Siparişleri
KU32S07	004	Group			Yazdırma İşlemleri
KU32P11	005	Display	KU32M05		Başlık Ekranı
KU32P12	005	Display	KU32M06		Detay Ekranı

Ek 3.16. Sağlama Servisi Sipariş İzleme Uygulama Yapısı (KU32A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU33P01	001	Execute			Execute KU33A
KU33P02	002	Converse	KU33M01		Arama Seçenekleri
KU33P03	003	Converse	KU33M02		Bilginin Girilmesi
KU33P04	004	Setinq	KU33R01	EZERTN	ISBN No Erişim
KU33P05	004	Setinq	KU33R01	EZERTN	Yazar Adı'ndan Erişim
KU33P06	004	Setinq	KU33R01	EZERTN	Kitap Adı'ndan Erişim
KU33P07	004	Setinq	KU33R01	EZERTN	Yayın Yılı'ndan Erişim
KU33P08	004	Scan	KU33R01	EZERTN	Scan KU33R01
KU33S01	004	Group			Kayıt Working Storage'a
KU33P08	005	Scan	KU33R01	EZERTN	Scan KU33R01
KU33P09	004	Converse	KU33M03		Sipariş Bilgileri Ekrana
KU33S02	005	Group			Working Storage'ı Ekrana Taşı
KU33S03	005	Group			Ekranı Working Storage'a Taşı
KU33S04	005	Group			Working Storage'ı Boşalt
KU33S05	006	Group			Kursorun Tesbiti
KU33S06	007	Group			Kursor Bulundu
KU33S07	007	Group			Başlangıç Bilgileri
KU33P10	008	Inquiry	KU33R02	EZERTN	Sıra Numarasını Bul
KU33P11	008	Setinq	KU33R02	EZERTN	Setinq KU33R02
KU33P12	008	Scan	KU33R02	EZERTN	Scan KU33R02
KU33P13	008	Converse	KU33M04		Muhasebe Bilgileri Ekrana
KU33S08	009	Group			Working Storage'ı Ekrana Taşı
KU33S09	009	Group			Ekranı Working Storage'a Taşı
KU33S10	009	Group			Kayıt - Silme - Değişiklik mi ?
KU33S11	010	Group			İşlemlere Başlangıç
KU33P14	010	Add	KU33R02	EZERTN	Kayıt
KU33P15	010	Sqlxec	KU33R02	EZERTN	Silme
KU33P16	010	Sqlxec	KU33R02	EZERTN	Değişiklik
KU33S12	009	Group			Muhasebe Bilgileri Yazıcıya
KU33P17	010	Display	KU33M06		Başlık Ekranı
KU33P18	010	Display	KU33M07		Detay Ekranı
KU33P19	010	Display	KU33M08		Toplam Ekranı
KU33P20	008	Converse	KU33M05		Yayın Kitap İse
KU33P21	009	Inquiry	KU33R01	EZERTN	ISBN No'sunu Sorgula

Ek 3.17. Sağlama Servisi Muhasebe İşlemleri Uygulama Yapısı (KU33A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU34P01	001	Execute			Execute KU34A
KU34P02	002	Converse	KU34M01		Fatura Seçenekleri
KU34P03	003	Converse	KU34M02		Bilginin Girilmesi
KU34P04	004	Setinq	KU34R01	EZERTN	Sale Invoice Setinq
KU34P05	004	Setinq	KU34R01	EZERTN	Extra Charge Setinq
KU34P06	004	Setinq	KU34R01	EZERTN	Credit Note Setinq
KU34P07	004	Scan	KU34R01	EZERTN	Scan KU34R01
KU34S01	004	Group			Kayıt Working Storage'a
KU34P08	005	Inquiry	KU34R02	EZERTN	ISBN No'sundan Yayının
KU34P07	005	Scan	KU34R01	EZERTN	Scan KU34R01
KU34P09	005	Converse	KU34M03		Detaylar Ekrana
KU34S02	006	Group			Working Storage'ı Ekrana Taşı
KU34S03	006	Group			Ekranı Working Storage'a Taşı
KU34S04	006	Group			Listeleme
KU34P10	007	Display	KU34M04		Başlık Ekranı
KU34P11	007	Display	KU34M05		Detay Ekranı
KU34P12	007	Display	KU34M06		Fatura Toplam Ekranı
KU34P13	007	Display	KU34M07		Ayrıntılar Ekranı

Ek 3.18. Sağlama Servisi Faturaların Takibi Uygulama Yapısı (KU34A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU35P01	001	Execute			Execute KU35A
KU35P02	002	Converse	KU35M01		Fatura Dağılım Ekranı
KU35S01	003	Group			Seçenekler
KU35P03	004	Inquiry	KU35R01	EZERTN	Sale Invoice Inquiry
KU35P04	004	Inquiry	KU35R01	EZERTN	Extra Charge Inquiry
KU35P05	004	Inquiry	KU35R01	EZERTN	Credit Note Inquiry
KU35P06	004	Converse	KU35M02		Detaylar Ekrana
KU35P07	005	Display	KU35M03		Ekran Bilgileri Yazıcıya
KU35P08	005	Display	KU35M04		Detay Ekranı
KU35P09	005	Display	KU35M05		Son Satır
KU35P10	004	Setinq	KU35R02	EZERTN	Kitaplara Ait Fatura Dağılımı
KU35S11	004	Scan	KU35R02	EZERTN	Scan KU35R02

Ek 3.19. Sağlama Servisi Faturaların Dağılımı Uygulama Yapısı (KU35A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU36P01	001	Execute			Execute KU36A
KU36P02	002	Converse	KU36M01		Firma Bazında Dağılım
KU36P03	003	Setinq	KU36R01	EZERTN	Firma Bazında Dağı. Tesbiti
KU36P04	003	Scan	KU36R01	EZERTN	Scan KU36R01

Ek 3.20. Sağlama Servisi Firma Bazında Dağılım Uygulama Yapısı (KU36A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU37P01	001	Execute			Execute KU37A
KU37P02	002	Converse	KU37M01		Listeleme Seçimi
KU37S01	003	Group			Listeleme Yayın Adına Göre
KU37P03	004	Setinq	KU37R01	EZERTN	Yayın Adına Göre Setinq
KU37P04	004	Scan	KU37R01	EZERTN	Scan KU37R01
KU37P05	004	Display	KU37M02		Başlık Ekranı
KU37P06	004	Display	KU37M03		Detay Ekranı
KU37P09	004	Display	KU37M04		Son Satır
KU37S02	003	Group			Listeleme Firma Adına Göre
KU37P07	004	Setinq	KU37R01	EZERTN	Firma Adına Göre Setinq
KU37P08	004	Scan	KU37R01	EZERTN	Scan KU37R01
KU37P05	004	Display	KU37M02		Başlık Ekranı
KU37P09	004	Display	KU37M04		Son Satır
KU37P06	004	Display	KU37M03		Detay Ekranı

Ek 3.21. Sağlama Servisi Yayın Listeleme Uygulama Yapısı (KU37A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU40P01	001	Execute			Execute KU40A
KU40P02	002	Converse	KU40M01		Üyelik İşlemleri
KU40S01	003	Group			Başlangıç İşlemleri
KU40P03	004	Inquiry	KU40R01	EZERTN	Kart No Inquiry
KU40S02	003	Group			Kontrollerin Yapılması
KU40P04	004	Add	KU40R01	EZERTN	Kayıt Process'İ
KU40S03	003	Group			Kayıt İşlemleri
KU40P05	004	Sqlexec	KU40R01	EZERTN	Değişiklik Process'i
KU40S04	003	Group			Değişiklik İşlemleri
KU40P06	004	Sqlexec	KU40R01	EZERTN	Uyeler SQL Tablosundan Sil
KU40P07	004	Sqlexec	KU40R02	EZERTN	Odunc SQL Tablosundan Sil
KU40S05	003	Group			Silme İşlemleri
KU40P08	004	Inquiry	KU40R01	EZERTN	Kart No'sundan Arama
KU40P09	004	Inquiry	KU40R01	EZERTN	Sicil No'dan Arama

Ek 3.22. Ödünç Verme Servisi Üyelik İşlemleri Uygulama Yapısı (KU40A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU41P01	001	Execute			Execute KU41A
KU41P02	002	Converse	KU41M01		Diğer Bilgiler Ekrana
KU41P03	003	Setinq	KU41R01	EZERTN	Setinq Ad - Soyad
KU41P04	003	Scan	KU41R01	EZERTN	Scan KU41R01
KU41S01	003	Group			Scan Sonrası Atamalar
KU41P04	004	Scan	KU41R01	EZERTN	Scan KU41R01
KU41P05	003	Converse	KU41M02		Ad - Soyad Sorgulama
KU41S02	004	Group			Working Storage'ı Ekrana Taşı
KU41S03	004	Group			Ekranı Working Storage'a Taşı
KU41S04	004	Group			Kursurun Tesbiti
KU41S05	005	Group			Kursor Bulundu

Ek 3.23. Öd. Ver. Ser. Ödünç Verme - Geri Alma Uygulama Yapısı (KU41A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU42P01	001	Execute			Execute KU42A
KU42P02	002	Converse	KU42M01		Ödünç Verme İşlemleri
KU42S01	003	Group			Başlangıç İşlemleri
KU42P03	004	Inquiry	KU42R02	EZERTN	Maksimum Sıra No'su Tesbit
KU42S02	003	Group			Kontroller
KU42P04	004	Inquiry	KU42R01	EZERTN	Kart No'suna Göre Arama
KU42S03	003	Group			Tarih Bilgisi Taşınıyor
KU42S04	003	Group			Atamalar & Add & Update
KU42P05	004	Add	KU42R02	EZERTN	Kayıt İşlemleri
KU42P06	004	Sqlxec	KU42R01	EZERTN	Alış Sayısını Güncelleştir

Ek 3.24. Ödünç Verme Servisi Ödünç Verme Uygulama Yapısı (KU42A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU43P01	001	Execute			Execute KU43A
KU43S01	002	Group			Başlangıç İşlemleri
KU43S02	002	Group			Arama Başlangıcı
KU43P02	003	Inquiry	KU43R01	EZERTN	Kart No Inquiry
KU43P03	003	Setinq	KU43R02	EZERTN	Verilen Yayınlar
KU43P04	003	Scan	KU43R02	EZERTN	Scan KU43R02
KU43S03	003	Group			Kayıt Bilgileri Working Stor.'a
KU43P04	004	Scan	KU43R02	EZERTN	Scan KU43R02
KU43P05	003	Converse	KU43M01		Bilgilerin Taranması
KU43S04	004	Group			Working Storage'ı Ekrana Taşı
KU43S05	004	Group			Ekranı Working Storage'a Taşı
KU43S06	004	Group			Tarih Bilgileri Taşınıyor
KU43P06	005	Sqlxec	KU43R03	EZERTN	Güncelleştirme
KU43S07	004	Group			Yazıcıdan Bilgilerin Alınması
KU43P07	005	Display	KU43M02		Başlık Ekranı
KU43P08	005	Display	KU43M03		Detay Ekranı
KU43P09	005	Display	KU43M04		Ara Çizgilerin Bastırılması

Ek 3.25. Ödünç Verme Servisi Yayın İade Uygulama Yapısı (KU43A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU44P01	001	Execute			Execute KU44A
KU44S01	002	Group			Başlangıç İşlemleri
KU44P02	003	Setinq	KU44R02	EZERTN	Kitap Alış Tarihi >= 15
KU44P03	003	Scan	KU44R02	EZERTN	Scan KU44R02
KU44S02	003	Group			Kayıt Bilgileri Working Stor.'a
KU44P03	004	Scan	KU44R02	EZERTN	Scan KU44R02
KU44P04	003	Converse	KU44M01		Süresini Aşan Şahıslar Listesi
KU44S03	004	Group			Working Storage'ı Ekrana Taşı
KU44S04	004	Group			Ekranı Working Storage'a Taşı
KU44S05	004	Group			Kursurun Tesbiti
KU44S06	005	Group			Kursor Bulundu
KU44P05	006	Inquiry	KU44R01	EZERTN	Inquiry KU44R01

Ek 3.26. Ödünç Verme Servisi Süresini Aşan Şahıslar Uygulama Yapısı (KU44A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU45P01	001	Execute			Execute KU45A
KU45P02	002	Converse	KU45M01		Arama Bilgilerini Gir
KU45S01	003	Group			Başlangıç İşlemleri
KU45P03	003	Setinq	KU45R01	EZERTN	Kitap Adından Şahsı Bul
KU45P04	003	Setinq	KU45R01	EZERTN	Yıl, Cilt, Sayı'dan Erişim
KU45P05	003	Scan	KU45R01	EZERTN	Scan KU45R01
KU45S02	003	Group			Kayıt Bilgileri Working Stor.'a
KU45P05	004	Scan	KU45R01	EZERTN	Scan KU45R01
KU45P06	004	Converse	KU45M02		Tarama Sonuçları Ekrana
KU45S03	005	Group			Working Storage'ı Ekrana Taşı
KU45S04	005	Group			Ekranı Working Storage'a Taşı
KU45S05	005	Group			Kursorun Tesbiti
KU45S06	006	Group			Kursor Bulundu
KU45P07	007	Inquiry	KU45R01	EZERTN	Kart No'sundan Sorgulama

Ek 3.27. Ödünç Verme Servisi Yayının Kimde Olduğu Uygulama Yapısı (KU45A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU46P01	001	Execute			Execute KU46A
KU46P02	002	Converse	KU46M01		Kullanıcı Gupları Dağılımı
KU46P03	003	Setinq	KU46R01	EZERTN	Ünvana Göre Dağılım
KU46P04	003	Scan	KU46R01	EZERTN	Scan KU46R01

Ek 3.28. Ödünç Verme Servisi Kullanıcı Dağılımı Uygulama Yapısı (KU46A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU47P01	001	Execute			Execute KU47A
KU47P02	002	Converse	KU47M01		Converse KU47M01
KU47P03	003	Inquiry	KU47R01	EZERTN	Aylık - Yıllık Toplam
KU47P04	003	Inquiry	KU47R01	EZERTN	Aylık - Yıllık Toplam
KU47P05	003	Inquiry	KU47R01	EZERTN	Aylık - Yıllık Toplam

Ek 3.29. Ödünç Verme Servisi Aylık-Yıllık İşlem Akışı Uygulama Yapısı (KU47A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU51P01	001	Execute			Execute KU51A
KU51P02	002	Converse	KU51M01		1 / 25.000 Ölçekli Haritalar
KU51S01	003	Group			Başlangıç Bilgileri
KU51P03	004	Inquiry	KU51R01	EZERTN	Kayıtlı Harita Adedini Bul
KU51P04	004	Inquiry	KU51R01	EZERTN	Son Olarak İşlenen Bilgi
KU51P08	004	Setinq	KU51R01	EZERTN	Haritaların Seçimi
KU51P09	004	Scan	KU51R01	EZERTN	Scan KU51R01
KU51S02	003	Group			Working Storage Ekranı
KU51S03	003	Group			Ekran Bilgileri Work. Stor.'a
KU51S04	003	Group			İşlem K - D veya S mi ?
KU51S05	004	Group			Kayıt İşlemi İçin Atamalar
KU51P05	005	Add	KU51R01	EZERTN	Kayıt Process'i
KU51S06	004	Group			Değişiklik İşlemi Atamaları
KU51P06	005	Sqlexec	KU51R01	EZERTN	Değişiklik Process'i
KU51S07	004	Group			Silme İşlemi İçin Atamalar
KU51P07	005	Sqlexec	KU51R01	EZERTN	Silme Process'i
KU51S08	003	Group			Harita Bilgilerini Yazıcıya
KU51P08	004	Setinq	KU51R01	EZERTN	Haritaların Seçimi
KU51P09	004	Scan	KU51R01	EZERTN	Scan KU51R01
KU51P10	004	Display	KU51M02		Başlık Ekranı
KU51P11	004	Display	KU51M03		Detay Ekranı
KU51P12	004	Display	KU51M04		Son Bilgi Ekranı

Ek 3.30. Dökümantasyon Servisi 1/25.000'lik Harita Uygulama Yapısı (KU51A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU52P01	001	Execute			Execute KU52A
KU52P02	002	Converse	KU52M01		1 / 100.000 Ölçekli Haritalar
KU52S01	003	Group			Başlangıç Bilgileri
KU52P03	004	Inquiry	KU52R01	EZERTN	Kayıtlı Harita Adedini Bul
KU52P04	004	Inquiry	KU52R01	EZERTN	Son Olarak İşlenen Harita Adını
KU52P05	004	Setinq	KU52R01	EZERTN	Haritaların Seçimi
KU52P06	004	Scan	KU52R01	EZERTN	Scan KU52R01
KU52S02	003	Group			Working Storage Ekranı
KU52S03	003	Group			Ekran Bilgileri Working Stor.'a
KU52S04	003	Group			İşlem K - D veya S mi ?
KU52S05	004	Group			Kayıt İşlemi İçin Atamalar
KU52P07	005	Add	KU52R01	EZERTN	Kayıt process'i
KU52S06	004	Group			Değişiklik İşlemi İçin Atamalar
KU52P08	005	Sqlexec	KU52R01	EZERTN	Değişiklik Process'i
KU52S07	004	Group			Silme İşlemi İçin Atamalar
KU52P09	005	Sqlexec	KU52R01	EZERTN	Silme Process'i
KU52S08	003	Group			Harita Bilgileri Yazıcıya
KU52P05	004	Setinq	KU52R01	EZERTN	Haritaların Seçimi
KU52P06	004	Scan	KU52R01	EZERTN	Scan KU52R01
KU52P10	004	Display	KU52M02		Başlık Ekranı
KU52P11	004	Display	KU52M03		Detay Ekranı
KU52P12	004	Display	KU52M04		Son Bilgi Ekranı
KU52P13	004	Display	KU52M05		Boş Satır İlave

Ek 3.31. Dökümantasyon Servisi 1/100.000'lik Harita Uygulama Yapısı (KU52A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU53P01	001	Execute			Execute KU53A
KU53P02	002	Converse	KU53M01		1 / 250.000 Ölçekli Haritalar
KU53S01	003	Group			Başlangıç Bilgileri
KU53P03	004	Inquiry	KU53R01	EZERTN	Kayıtlı Harita Adedini Bul
KU53P04	004	Inquiry	KU53R01	EZERTN	Son Olarak İşlenen Harita
KU53P08	004	Setinq	KU53R01	EZERTN	Haritaların Seçimi
KU53P09	004	Scan	KU53R01	EZERTN	Scan KU53R01
KU53S02	003	Group			Working Storage Ekrana
KU53S03	003	Group			Ekran Bilgileri Work. Stor.'a
KU53S04	003	Group			İşlem K - D veya S mi ?
KU53S05	004	Group			Kayıt İşlemi İçin Atamalar
KU53P05	005	Add	KU53R01	EZERTN	Kayıt Process'i
KU53S06	004	Group			Değişiklik İşlemi Atamalar
KU53P06	005	Sqlxec	KU53R01	EZERTN	Değişiklik Process'i
KU53S07	004	Group			Silme İşlemi İçin Atamalar
KU53P07	005	Sqlxec	KU53R01	EZERTN	Silme Process'i
KU53S08	003	Group			Harita Bilgileri Yazıcıya
KU53P08	004	Setinq	KU53R01	EZERTN	Haritaların Seçimi
KU53P09	004	Scan	KU53R01	EZERTN	Scan KU53R01
KU53P10	004	Display	KU53M02		Başlık Ekranı
KU53P11	004	Display	KU53M03		Detay Ekranı
KU53P12	004	Display	KU53M04		Son Bilgi Ekranı

Ek 3.32. Dökümantasyon Servisi 1/250.000'lik Harita Uygulama Yapısı (KU53A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
KU54P01	001	Execute			Execute KU54A
KU54P02	002	Converse	KU54M01		Hava Fotoğrafları
KU54S01	003	Group			Başlangıç Bilgileri
KU54P03	004	Inquiry	KU54R01	EZERTN	Kayıtlı Fotoğraf Adedini Bul
KU54P04	004	Inquiry	KU54R01	EZERTN	Son Olarak İşlenen Not
KU54P08	004	Setinq	KU54R01	EZERTN	Hava Fotoğraflarının Seçimi
KU54P09	004	Scan	KU54R01	EZERTN	Scan KU54R01
KU54S02	003	Group			Working Storage Ekrana
KU54S03	003	Group			Ekran Bilgileri Work. Stor.'a
KU54S04	003	Group			İşlem K - D veya S mi ?
KU54S05	004	Group			Kayıt İşlemi İçin Atamalar
KU54P05	005	Add	KU54R01	EZERTN	Kayıt Process'i
KU54S06	004	Group			Değişiklik İşlemi Atamalar
KU54P06	005	Sqlxec	KU54R01	EZERTN	Değişiklik Process'i
KU54S07	004	Group			Silme İşlemi İçin Atamalar
KU54P07	005	Sqlxec	KU54R01	EZERTN	Silme Process'i
KU54S08	003	Group			Fotoğraf Bilgileri Yazıcıya
KU54P08	004	Setinq	KU54R01	EZERTN	Fotoğrafların Seçimi
KU54P09	004	Scan	KU54R01	EZERTN	Scan KU54R01
KU54P10	004	Display	KU54M02		Başlık Ekranı
KU54P11	004	Display	KU54M03		Detay Ekranı
KU54P12	004	Display	KU54M04		Son Bilgi Ekranı

Ek 3.33. Dökümantasyon Servisi Hava Fotoğrafları Uygulama Yapısı (KU54A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA10P01	001	Execute			Tarama İşlemlerinin Seçimi
TA10P02	002	Converse	TA10M01		Tarama İşlemlerini Görüntüle

Ek 3.34. Yayın Tarama İşlemleri Uygulama Yapısı (TA10A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA20P01	001	Execute			Execute TA20A
TA20P02	002	Converse	TA20M01		Kitap Tarama Seçenekleri

Ek 3.35. Kitap Tarama İşlemleri Uygulama Yapısı (TA20A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA30P01	001	Execute			Execute TA30A
TA30P02	002	Converse	TA30M01		Sürekli Yayın Tarama Seçenekleri

Ek 3.36. Sürekli Yayın Tarama İşlemleri Uygulama Yapısı (TA30A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA40P01	001	Execute			Execute TA40A
TA40P02	002	Converse	TA40M01		Tez - Makale Tarama Seç.

Ek 3.37. Tez - Makale Tarama İşlemleri Uygulama Yapısı (TA40A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA50P01	001	Execute			Execute TA50A
TA50P02	002	Converse	TA50M01		Harita Tarama Seçenekleri

Ek 3.38. Harita Tarama İşlemleri Uygulama Yapısı (TA50A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA21P01	001	Execute			Execute TA21A
TA21P02	002	Converse	TA21M01		Tarama Seçeneğinin Girilmesi
TA21S00	003	Group			Enter Tuşuna Basılmışsa
TA21S01	004	Group			Seçime Bağlı Process'leri
TA21P04	005	Setinq	TA21R01	EZERTN	Kitap Adı Tarama
TA21P05	005	Setinq	TA21R01	EZERTN	Yazar Adı Tarama
TA21P06	005	Setinq	TA21R01	EZERTN	Konu No Tarama
TA21P07	005	Setinq	TA21R01	EZERTN	Konu Adı Tarama
TA21P08	005	Setinq	TA21R01	EZERTN	Seri Adı Tarama
TA21P09	005	Setinq	TA21R01	EZERTN	Emeği Geçenleri Tarama
TA21P10	005	Setinq	TA21R01	EZERTN	ISBN No Tarama
TA21P11	005	Setinq	TA21R01	EZERTN	Yayın Yılı Tarama
TA21P12	005	Setinq	TA21R01	EZERTN	Yayın Dili Tarama
TA21P13	005	Scan	TA21R01	EZERTN	Scan TA21R01
TA21P03	005	Converse	TA21M02		Bilgiler Ekrana
TA21S03	006	Group			Working Storage'dan Ekrana
TA21S04	006	Group			Ekrandan Working Storage'a
TA21S05	006	Group			Kursorun Tesbiti
TA21S06	007	Group			Kursor Bulundu
TA21P14	008	Inquiry	TA21R01	EZERTN	Ayrıntılı Bilgi
TA21S07	008	Group			Kitap Bilgilerini Bul
TA21S08	009	Group			Bilgiler Working Storage'a
TA21S02	010	Group			Kayıt Working Storage'a
TA21S09	009	Group			Bilgi Son Noktalarının Tesbiti
TA21S10	009	Group			Bilgiler Yapıya
TA21P15	009	Converse	TA21M03		Ayrıntılı Bilgi Ekranı
TA21S11	006	Group			Bilgilerin Yazıcıdan Alınması
TA21P14	007	Inquiry	TA21R01	EZERTN	Ayrıntılı Bilgiler
TA21S08	007	Group			Bilgiler Working Storage'a
TA21S02	008	Group			Kayıt Working Storage'a
TA21S09	007	Group			Bilgi Son Noktalarının Tesbiti
TA21S10	007	Group			Bilgiler Yapıya
TA21P16	007	Display	TA21M04		Konu no, Yazar ve Ayırıcı Kod
TA21P17	007	Display	TA21M05		Ayrıntılar Yazıcıya
TA21P18	007	Display	TA21M06		Çizgilerin Çizilmesi

Ek 3.39. Kitap Tarama İşlemleri Uygulama Yapısı (TA21A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA3IP01	001	Execute			Execute TA31A
TA3IP02	002	Converse	TA3IM01		Tarama Seçeneğinin Girilmesi
TA3IS00	003	Group			Bilgilerin Taşınması
TA3IS01	004	Group			Seçime Bağlı Process'leri
TA3IS02	005	Group			Büyük Harfe Çevir
TA3IP03	005	Setinq	TA3IR01	EZERTN	Sürelî Yayın Adından Tarama
TA3IP04	005	Setinq	TA3IR01	EZERTN	Yayın Yılı Tarama
TA3IP05	005	Setinq	TA3IR01	EZERTN	Yayınlayan Tarama
TA3IP06	005	Setinq	TA3IR01	EZERTN	ISSN No Tarama
TA3IP07	005	Scan	TA3IR01	EZERTN	Scan TA3IR01
TA3IP08	005	Converse	TA3IM02		Bilgiler Ekrana
TA3IS03	006	Group			Working Storage'dan Ekrana
TA3IS04	006	Group			Ekrandan Working Storage'a
TA3IS05	006	Group			Kursurun Tesbiti
TA3IS06	007	Group			Kursor Bulundu
TA3IP09	008	Inquiry	TA3IR01	EZERTN	Sipariş No'dan Erişim
TA3IS07	008	Group			Dönüştürme Programları
TA3IS08	009	Group			Kayıt Working Storage'a
TA3IS09	010	Group			Son Noktaların Tesbiti
TA3IS10	009	Group			Küçük Harfe Çevir
TA3IP10	009	Converse	TA3IM03		Ayrıntılı Bilgi Ekranı
TA3IS11	006	Group			Bilgilerin Yazıcıdan Alınması
TA3IP14	007	Inquiry	TA3IR01	EZERTN	Sipariş No Inquiry
TA3IS08	007	Group			Kayıt Working Storage'a
TA3IS09	008	Group			Son Noktaların Tesbiti
TA3IS091	007	Group			Küçük Harfe Çevir
TA3IS10	007	Group			Ekrana veya Yazıcıya
TA3IP11	007	Display	TA3IM04		Detay Satırları
TA3IP12	007	Display	TA3IM05		Çizgi Çiz

Ek 3.40. Sürelî Yayın Tarama İşlemleri Uygulama Yapısı (TA31A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA41P01	001	Execute			Execute TA41A
TA41P02	002	Converse	TA41M01		Tarama Seçeneğinin Girilmesi
TA41S00	003	Group			Enter Tuşuna Basılmışsa
TA41S01	004	Group			Seçime Bağlı Process'leri
TA41P04	005	Setinq	TA41R01	EZERTN	Tez-Makale Adı Tarama
TA41P05	005	Setinq	TA41R01	EZERTN	Tez-Makale Yazar Adı Tarama
TA41P06	005	Setinq	TA41R01	EZERTN	Tez-Makale Konu No Tarama
TA41P07	005	Setinq	TA41R01	EZERTN	Tez-Makale Konu Adı Tarama
TA41P08	005	Setinq	TA41R01	EZERTN	Tez-Makale Seri Adı Tarama
TA41P11	005	Setinq	TA41R01	EZERTN	Tez-Makale Yayın Yılı Tarama
TA41P12	005	Setinq	TA41R01	EZERTN	Tez-Makale Yayın Dili Tarama
TA41P13	005	Scan	TA41R01	EZERTN	Scan TA41R01
TA41P03	005	Converse	TA41M02		Bilgiler Ekrana
TA41S03	006	Group			Working Storage'dan Ekrana
TA41S04	006	Group			Ekrandan Working Storage'a
TA41S05	006	Group			Kursurun Tesbiti
TA41S06	007	Group			Kursor Bulundu
TA41P14	008	Inquiry	TA41R01	EZERTN	Ayrıntılı Bilgi
TA41S07	008	Group			Kitap Bilgilerini Bul
TA41S08	009	Group			Bilgiler Working Storage'a
TA41S02	010	Group			Kayıt Working Storage'a
TA41S09	009	Group			Bilgi Son Noktalarının Tesbiti
TA41S10	009	Group			Bilgiler Yapıya
TA41P15	009	Converse	TA41M03		Ayrıntılı Bilgi Ekranı
TA41S11	006	Group			Bilgilerin Yazıcıdan Alınması
TA41P14	007	Inquiry	TA41R01	EZERTN	Ayrıntılı Bilgiler
TA41S08	007	Group			Bilgiler Working Storage'a
TA41S02	007	Group			Kayıt Working Storage'a
TA41S09	007	Group			Bilgi Son Noktalarının Tesbiti
TA41S10	007	Group			Bilgiler Yapıya
TA41P16	007	Display	TA41M04		Konu no, Yazar ve Ayrırcı Kod
TA41P17	007	Display	TA41M05		Ayrıntılar Yazıcıya
TA41P18	007	Display	TA41M06		Çizgilerin Çizilmesi

Ek 3.41. Tez - Makale Tarama İşlemleri Uygulama Yapısı (TA41A)

NAME	LVL	OPTION	OBJECT	ERROR	DESCRIPTION
TA51P01	001	Execute			Execute TA51A
TA51P02	002	Converse	TA51M01		Tarama Seçeneğinin Girilmesi
TA51S01	003	Group			Kontroller
TA51P03	004	Setinq	TA51R01	EZERTN	Harita Ölçeği 1/ 25.000
TA51P04	004	Scan	TA51R01	EZERTN	Scan TA51R01
TA51P05	004	Setinq	TA51R02	EZERTN	Harita Ölçeği 1/ 100.000
TA51P06	004	Scan	TA51R02	EZERTN	Scan TA51R02
TA51P07	004	Setinq	TA51R03	EZERTN	Harita Ölçeği 1/ 250.000
TA51P08	004	Scan	TA51R03	EZERTN	Scan TA51R03
TA51P09	004	Setinq	TA51R04	EZERTN	Hava Fotoğrafları
TA51P10	004	Scan	TA51R04	EZERTN	Scan TA51R04
TA51P11	004	Converse	TA51M02		Harita Bilgileri Ekrana
TA51S02	005	Group			Working Storage'dan Ekrana
TA51S03	005	Group			Ekrandan Working Storage'a
TA51S04	005	Group			Bilgileri Yazıcıya Gönder
TA51P12	006	Display	TA51M03		Başlık Ekranı
TA51P13	006	Display	TA51M04		Detay Ekranı

Ek 3.42. Harita Bilgileri Tarama İşlemleri Uygulama Yapısı (TA51A)

Ek 4. Uygulamaya Ait Ekran Panelleri

KU00M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ana Menü
1- Katalog ve Sınıflandırma Servisi 2- Süreli Yayınlar Servisi 3- Sağlama Servisi 4- Üyelik ve Ödünç Verme Servisi 5- Dökümantasyon Servisi 6- Çıkış	
Tercihinizi Giriniz :__	
Enter : İşlem	

Ek 4.1. Ana Menü paneli

KU01M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama ve Sınıflandırma Servisi
1- Kayıt İşlemleri 2- Değişiklik - Silme İşlemleri 3- Kataloglama Seçenekleri 4- Çıkış	
Tercihinizi Giriniz :__	
Enter : İşlem	

Ek 4.2. Kataloglama Servisi İşlem Seçenekleri

KU02M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayınlar Servisi

- 1- Sürelî Yayın Takibi
- 2- Sürelî Yayın Kataloğu
- 3- Sürelî Yayın Katalog Kartları
- 4- Çıkış

Tercihinizi Giriniz : __

Enter : İşlem

Ek 4.3. Sürelî Yayınlar Servisi İşlem Seçenekleri

KU03M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1- Sipariş Bilgi Girişi
- 2- Sipariş Düzeltme - Silme
- 3- Siparişlerin Takibi
- 4- Muhasebe İşlemleri
- 5- Faturaların Takibi
- 6- Parasal Dağılım
- 7- Firma Bazında Dağılım
- 8- Listeler (Alfabetik ve Firma Bazında)
- 9- Çıkış

Tercihinizi Giriniz : __

Enter : İşlem

Ek 4.4. Sağlama Servisi İşlem Seçenekleri

KU04M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Ödünç Verme Servisi

- 1- Üye Kayıtların Yapılması
- 2- Ödünç Verme / Geri Alma İşlemleri
- 3- Süresi Dolan Kitap ve Dergiler
- 4- Yayınların Kimde Olduğunu Görme
- 5- Kullanıcı Gruplarına Göre Dağılım
- 6- Aylık / Yıllık İşlem Durumu
- 7- Çıkış

Tercihinizi Giriniz :__

Enter : İşlem

Ek 4.5. Ödünç Verme Servisi İşlem Seçenekleri

KU05M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Dökümantasyon Servisi

- 1- 1/25.000 Ölçekli Haritalar
- 2- 1/100.000 Ölçekli Haritalar
- 3- 1/250.000 Ölçekli Haritalar
- 4- Hava Fotoğrafları
- 5- Çıkış

Tercihinizi Giriniz :__

Enter : İşlem

Ek 4.6. Dökümantasyon Servisi İşlem Seçenekleri

KU10M01		ATATÜRK ÜNİVERSİTESİ	
		Merkez Kütüphanesi	
		Kataloglama ve Sınıflandırma Servisi	
Belge Türü :		Tez Kodu :	
Demirbaş No :		Yazar Kodu :	Ayırıcı Kod :
Konu No :			
ISBN No :			
Yayın Adı :			
Yazar Adı :		Yazar Soyadı :	
Emeği Geç :			
Yayın Yeri :			
Yayınlayan :			
Yayın Yılı :			
Adedi :		Yayın Konumu :	
PF3 : Çıkış		PF4 : Kayıt	PF8 : İleri

Ek 4.7. Kataloglama Servisi Bilgi Giriş Paneli-1

KU10M02		ATATÜRK ÜNİVERSİTESİ	
		Merkez Kütüphanesi	
		Kataloglama ve Sınıflandırma Servisi	
Seri (DİZİ) Adı :			
Konu Başlıkları :			
Bibliyografya :			
Açıklamalar :			
Dizin Bilgisi :			
Bakı Kaydı :			
Sayfa ve Cilt :			
Fiziki Özellik. :			
Yayın Dili :		Kayıt Ortamı :	
Bu Yayına Daha Önceden Verilmiş Demirbaş Numaraları :			
İzleme Kaydı :			
PF3 : Çıkış		PF4 : Kayıt	PF7 : Geri

Ek 4.8. Kataloglama Servisi Bilgi Giriş Paneli-2

KU10H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Katalog Bilgilerine Ait Veri Girişi Yapılırken :

- 1- Veri Girişi sonrasında **PF4** ile kayıt yapmaya çalışmış ve bir hata mesajı almış iseniz bilgilerinizi gözden geçiriniz.
- 2- **KU10M01** panelinde iken **PF8** tuşuna basarak **KU10M02** paneline geçerek diğer bilgileri kodlayabilirsiniz.
- 3- Giriş Bilgileri içerisinde :
 Belge türü, dil, kayıt ortamı, yayın konumu bazı ön kabuller ile gelmektedir. İstenirse bu değerler değiştirilebilir. Değişiklikler yapılmaz ise ekranda görülen şekli ile kaydedilecektir.
- 4- Belge Türü **MU-NE-EY** ise bu yayın **DIŞARI ÇIKARILAMAZ** olarak kodlanır. (**Müracaat Eseri - Nadir Eserler - El Yazmaları**)
- 5- Belge türü, dil, kayıt ortamı, yayın konumu kodlarını görmek için **PF8** tuşunu kullanabilirsiniz.
- 6- Tez ve makale bilgileri de bu bölümden girilecektir. Tez kodlarını görmek için de **PF8** tuşunu kullanın.

PF3 : Çıkış**PF8 : Sayfa Çevir**

Ek 4.9. Kataloglama Servisi Bilgi Yardım Paneli-1

KU10H02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Belge Türü Kodları		Dil Kodları		Kayıt Ortamı Kodları	
KT	Kitap	TUR	Türkçe	NR	Normal Kağıt
TR	Teknik Rapor	ING	İngilizce	FL	Film
RS	Resim	ALM	Almanca	MF	MikroFilm/Fiş
EY	El Yazması	OSM	Osmanlıca	SL	Slayt
NE	Nadir Eserler	ARP	Arapça	CD	Compact Disc
MU	Müraacat Eseri	FRA	Fransızca	VI	Video
PT	Patent-Copyright	RUS	Rusça		
TZ	Tez				
MK	Makale				
		Tez Türü Kodları		Yayın Kodları	
		1-Yüksek Lisans		0- Kütühanede	
		2-Doktora Tezi		1- Okuyucuda	
		3-Tıpta Uzmanlık		2- Depoda	
		4-Doçentlik Tezi		3- Belli Değil	
		5-Bitirme Tezi			

PF3 : Çıkış**PF7 : Önceki Sayfa**

Ek 4.10. Kataloglama Servisi Yardım Paneli-2

KU11M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Kataloglama ve Sınıflandırma Servisi

- 1- ISBN No'dan Erişim
- 2- Yazar Adından Erişim
- 3- Yayın Adından Erişim
- 4- Çıkış

Tercihinizi Giriniz : 2

Enter : İşlem

Ek 4.11. Kataloglama Servisi Değişiklik-Silme Paneli

KU11M02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Kataloglama ve Sınıflandırma Servisi

Yazar adı :

PF3 : Çıkış

Enter : Arama

Ek 4.12. Kataloglama Servisi Değişiklik-Silme Bilgi Giriş Paneli

KU11M03		ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama ve Sınıflandırma Servisi	
		Sayfa :	
Yayın adı		Türü	
PF3 : Çıkış		PF4 : Ayrıntılar	
		PF7 : Geri	
		PF8 : İleri	

Ek 4.13. Kataloglama Servisi Değişiklik-Silme Gelen Bilgi Paneli

KU11M04		ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama ve Sınıflandırma Servisi	
Belge Türü :		Tez Kodu :	
Demirbaş No :		Yazar Kodu :	Ayrılcı Kod :
Konu No :			
ISBN No :			
Yayın Adı :			
Yazar Adı :		Yazar Soyadı :	
Emeđi Geç :			
Yayın Yeri :			
Yayınlayan :			
Yayın Yılı :			
Adedi :		Yayın Konumu :	
PF3 : Çıkış		PF5 : Deđişiklik	
		PF6 : Silme	
		PF8 : İleri	

Ek 4.14. Kataloglama Servisi Değişiklik-Silme Ayrıntılı Bilgi Paneli-1

KU11M05	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama ve Sınıflandırma Servisi		
Seri (DİZİ) Adı :			
Konu Başlıkları :			
Bibliyografya :			
Açıklamalar :			
Dizin Bilgisi :			
Bakı Kaydı :			
Sayfa ve Cilt :			
Fiziki Özellik :			
Yayın Dili :		Kayıt Ortamı :	
Bu Yayına Daha Önceden Verilmiş Demirbaş Numaraları :			
İzleme Kaydı :			
PF3 : Çıkış PF5 : Değişiklik PF6 : Silme PF7 : Geri			

Ek 4.15. Kataloglama Servisi Değişiklik-Silme Ayrıntılı Bilgi Paneli-2

KU11H01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri	
<u>Katalog Bilgilerine ait Veri Değişikliği ve Silme Yapılırken :</u>		
1- KU11M01 panelinde iken daha önceden girilen bir belgeye ISBN NO, yayın adı, yazar adı bilgilerinden erişebilir ve üzerinde PF5 ile değişiklik, PF6 ile silme yapabilirsiniz		
2- Örnek : Aranılan bilgi Katalog ise, bu bilgi aşağıdaki gibi girilmelidir.		
* Yayın adı Katalog ile başlayan yayınlara erişmek için	Katalog%	
* Yayın adı Katalog ile biten yayınlara erişmek için	%Katalog	
* Yayın adında Katalog Kelimesi geçen yayınlar için	%Katalog%	
3- Kayıt ortamı, yayın dili, kayıt türü, yayın konumu kodlarını görmek için PF8 tuşuna basınız.		
PF3 : Çıkış		PF7 : Sayfa Çevir

Ek 4.16. Kataloglama Servisi Değişiklik-Silme Yardım Paneli

KU12M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama ve Sınıflandırma Servisi
1- Temel Kart (Yazar Adı) 2- Kitap Adı 3- Konu Katalogu 4- Dizi Katalogu 5- Yer Katalogu 6- Çıkış	
Tercihinizi Giriniz : 2	
Enter : İşlem	

Ek 4.17. Kataloglama Servisi Katalog Kartı Bastırma Seçenekleri

KU13M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama ve Sınıflandırma Servisi
Yayın adını giriniz :	
PF3 : Çıkış Enter : Arama	

Ek 4.18. Kataloglama Servisi Katalog Kartı Bastırma Bilgi Giriş Paneli

KU13M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kataloglama					
Aranan :		Sayfa :				
Taraması yapılan bilgi ile alakalı olarak ____ adet yayın bulunmuştur.						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 70%;">Yayın Adı</th> <th style="width: 30%;">Türü</th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"></td> <td></td> </tr> </tbody> </table>		Yayın Adı	Türü			
Yayın Adı	Türü					
PF2 : Çıkış PF5 : Yazdır PF7 : Geri PF8 : İleri						

Ek 4.19. Kataloglama Servisi Katalog Kartı Bastırma Gelen Bilgi Paneli

KU20M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sürelî Yayınlar Servisi	
1- ISSN Numarasına Göre 2- Yayın Adına Göre 3- Firma Adına Göre 4- Çıkış		
Tercihinizi Giriniz : 2		
Enter : İşlem		

Ek 4.20. Sürelî Yayınlar Servisi Yayın Takip Seçenekleri

KU20M02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayınlar Servisi

Yayın adı :

PF3 : Çıkış

Enter : Arama

Ek 4.21. Sürelî Yayınlar Servisi Yayın Takip Bilgi Giriş Ekranı

KU20M03

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayınlar Servisi

Tarama yılı :

Sayfa :

Yayın adı

PF3 :Çıkış

PF4 : Ayrıntılar

PF7 : Geri

PF8 : İleri

PF9 :Yayın Takibi

Ek 4.22. Sürelî Yayınlar Servisi Yayın Takip Gelen Bilgi Ekranı

KU20M04	ATATÜRK ÜNİVERSİTESİ		
	Merkez Kütüphanesi		
	Sürelî Yayınlar Servisi		
ISSN No :			
İsmarlama Tar. :			
Dergi Adı :			
Volüm - Sayı :			
Fiziki Özellik. :			
Yayın Yeri :			
Yayın Tar. :	--/--/--		
Çıkış Ara. :			
Fak/Y.Ok./Ens. :			
Bölüm :			
Tarih - Sayı :	--/--/--	---	
İlk Geldiği Tar. :			
Satan Firma :			
Eski Adı -1 :			
Eski Adı -2 :			
Eski Adı -3 :			
Abone Dur. :	Yayın Dili	:	
PF3 : Çıkış			

Ek 4.23. Sürelî Yayınlar Servisi Yayın Takip Ayrıntılı Bilgi Paneli

KU20M05	ATATÜRK ÜNİVERSİTESİ																	
	Merkez Kütüphanesi																	
	Sürelî Yayınlar Servisi																	
	Sayfa :																	
Yayın Adı :																		
Firma Adı :																		
Çıkış Aralığı :																		
Yıl	Vol.	Sayı	Ay	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PF3 : Çıkış Enter : İşlem PF7 : Geri PF8 : İleri PF9 : Yazdır PF10 : Sağa																		

Ek 4.24. Sürelî Yayınlar Servisi Eksik Yayın Takip Paneli (Günlük ve Düzensiz)

KU20M06		ATATÜRK ÜNİVERSİTESİ															
		Merkez Kütüphanesi															
		Sürelî Yayınlar Servisi															
Yayın Adı :		Sayfa :															
Firma Adı :																	
Çıkış Aralığı :																	
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	İndeks	İşl.
PF3 : Çıkış Enter : İşlem PF7 : Geri PF8 : İleri PF11 : Sola																	

Ek 4.25. Sürelî Yayınlar Servisi Eksik Yayın Takip Paneli (Günlük ve Düzensiz)

KU20M07		ATATÜRK ÜNİVERSİTESİ															
		Merkez Kütüphanesi															
		Sürelî Yayınlar Servisi															
Yayın Adı :		Sayfa :															
Firma Adı :																	
Çıkış Aralığı :																	
Yıl	Vol.	Sayı	Oc.	Şub.	Mar	Nis.	May	Haz.	Tem.	Ağ.	Eyl.	Ek.	Kas.	Ara.	İndeks	İşlem	
PF3 : Çıkış Enter : İşlem PF7 : Geri PF8 : İleri PF9 : Yazdır																	

Ek 4.26. Sürelî Yayınlar Servisi Eksik Yayın Takip Paneli (Aylık)

KU20H01

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Yardım Bilgileri

Sürelî yayınlara ait yayın takibi yapılırken :

1- Bu bölümden Sürelî Yayınlar erişim 3 şekilde olmaktadır.

A- ISSN numarasından

B- Yayın adından

C- Firma adından

2- Örnek : Erişilecek yayının

* Adı Kitap ile başlıyorsa “%Kitap” şeklinde

* Adında Kitap sözcüğü geçiyorsa “%Kitap%” şeklinde

* Adının sonunda Kitap sözcüğü varsa “%Kitap” şeklinde

girilmesi gerekmektedir. Bu işlem ISSN No ve Firma adı taramalarında da geçerlidir.

PF3 : Çıkış

Ek 4.27. Sürelî Yayınlar Servisi Yardım Paneli-1

KU20H02

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Yardım Bilgileri

Sürelî yayınlara ait yayın takibi yapılırken :

1- Bu bölümden sürelî yayınlara erişim 3 şekilde olmaktadır.

A - ISSN numarasından

B - Yayın adından

C - Firma adından

2- Erişilen yayın adları ekrana gelmekte ve girildikten sonra, işlem hanesine :

Kayıt için K - Değişiklik için D - Silme için S harfi girip Enter tuşuna basılması yeterli olacaktır.

PF3 : Çıkış

Ek 4.28. Sürelî Yayınlar Servisi Yardım Paneli-2

T.C. YÜKSEKÖĞRETİM KURULU
T.C. MİLLÎ EĞİTİM BAKANLIĞI
T.C. KÜLTÜR, TURİZM VE
T.C. YATIRIM VE ENERJİ BAKANLIĞI
T.C. ÇEVRE, ŞEHİRCİLİK VE
T.C. İKLİM BAKANLIĞI
T.C. SAĞLIK BAKANLIĞI
T.C. SİVİL HAVACILIK GENEL MÜDÜRLÜĞÜ
T.C. ULUSLARARASI TİCARET BAKANLIĞI
T.C. YATIRIM VE ENERJİ BAKANLIĞI
T.C. ÇEVRE, ŞEHİRCİLİK VE
T.C. İKLİM BAKANLIĞI
T.C. SAĞLIK BAKANLIĞI
T.C. SİVİL HAVACILIK GENEL MÜDÜRLÜĞÜ
T.C. ULUSLARARASI TİCARET BAKANLIĞI

KU21M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayınlar Servisi

Listesini alacağınız sürelî yayının ilk harfini giriniz : _____
 (Tüm Liste için * giriniz.)

PF3 : Çıkış

Enter : İşlem

Ek 4.29. Sürelî Yayınlar Servisi Yayın Listesi Alma Paneli

KU21H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Bu program çalıştırıldığı zaman, Sürelî Yayınlar Katalogu'nun yazıcıdan alınmasını sağlar.
 Sürelî Yayınlar Katalogu'ndaki bilgiler şunlardır*.

*** AMERICAN JOURNAL OF PHYSICS**

(Sürelî Yayının Adı)

*** New York, 1933, 12**

(Yayın yeri, ilk yayınlandığı tarih, yayın aralığı bilgileri. Bu sahada -?- işareti varsa yayın yeri ve ilk yayına başladığı tarihin tesbit edilmediğini gösterir.)

*** 1968-1970, (1971), 1972 -**

(Kütüphaneye 1968 yılında gelmeye başlamış ve 1970 yılına kadar tam olarak gelmiş. 1971 yılı içinde eksik sayılar var. Satırın son bilgisi "-“ ise yayın gelmeye devam ediyor. “.” ise abonelik kesilmiş demektir.)

*** (EA : AMERICAN PHYSICS TEACHER 1933-1940)**

(Yayın 1933-1940 yılları arasında bu isimle çıkmış.)

***Kaynak :** Sürelî Yayınlar Katalogu, Atatürk üniversitesi Küt. ve Dök. D. Bşk., Yayın No:7, Erzurum, 1989, s. IV.

PF3 : Çıkış

Ek 4.30. Sürelî Yayınlar Servisi Yayın Listesi Alma Yardım Paneli

KU22M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayın Kartları

Yayın adını giriniz :

PF3 : Çıkış

Ek 4.31. Sürelî Yayınlar Servisi Sürelî Yayın Katalogu Bastırma Bilgi Giriş Paneli

KU22H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Bu ekranda Sürelî Yayınlar Katalog Kartı bilgileri bulunmaktadır.

Birinci Paragraf : Sürelî yayın adı, cilt-volüm-sayı-yıl, yayın yeri, yayınlayan, aboneliğin devam edip etmediği v.b. bilgiler yer almaktadır.

* İlk paragraftaki yıl bilgisinin sonunda ;

A) “,” varsa abonelik kesilmiş

B) “-“ varsa abonelik hala devam ediyor anlamına gelmektedir.

* Yayın yerinden önceki bilgi ;

A) “- — “ ise yayının çıkmakta olduğunu

B) “,” ise yayının kesildiğini göstermektedir.

İkinci Paragraf : Yayına ait sayfa-cilt bilgisi v.b. gibi fiziki özellikleri tanıttıcı bilgiler bulunmaktadır.

3. ve diğerleri : Paragraflar halinde çıkış periyodu, ISSN No, yayına ait eski isimleri ve açıklayıcı bilgiler vardır.

PF3 : Çıkış

Ek 4.32. Sürelî Yayınlar Servisi Yayın Katalogu Bastırma Yardım Paneli

KU30M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1- Kitap Siparişleri
 2- Dergi Siparişleri
 3- Çıkış

Tercihinizi Giriniz :

Enter : İşlem

Ek 4.33. Sağlama Servisi Sipariş İşlem Seçenekleri

KU30M02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

ISBN No	:				
İsmlama Tar.	:	--/--			
Yazarı	:				
Kitap Adı	:				
Cilt-Seri	:			Edisyon	:
Yayınlayan	:				
Yayın Yeri	:				
Yayın Tarihi	:			Yayın Dili	:
Fiati	:			Para Birimi	:
Fak./Y.Ok./Ens.	:				
Bölüm	:				
Tarih - Sayı	:	--/--	--		
Geldiği Tarih	:				
Satan	:				
Maliyeti	:				
Gel. Yapılacak İş.	:			Yer No	:
Fatura no	:			Demirbaş No	:

PF3 : Çıkış

PF4 : Kayıt

Ek 4.34. Sağlama Servisi Kitap Siparişi Bilgi Giriş Paneli

KU30M03	ATATÜRK ÜNİVERSİTESİ	
	Merkez Kütüphanesi	
	Sağlama Servisi	
ISSN No :	Önceki kayıt :	
İsmarlama Tar. :	--/--	
Dergi Adı :		
Volüm - Sayı :		
Fiziki Öz. :		
Yayın Yeri :		
Yayın Tarihi :		
Çıkış Ara. :		
Fak./Y.Ok./Ens. :		
Bölüm :		
Tarih - Sayı :	--/--	--
İlk Gel. Tarih :		
Satan Firma :		
Eski Adı-1 :		
Eski Adı-2 :		
Eski Adı-3 :		
Abone Durumu :	Yayın Dili :	
PF3 : Çıkış		PF4 : Kayıt

Ek 4.35. Sağlama Servisi Süreli Yayın Siparişi Bilgi Giriş Paneli

KU30H01	ATATÜRK ÜNİVERSİTESİ	
	Merkez Kütüphanesi	
	Yardım Bilgileri	
1-Sağlama Servisine ait siparişler 2 şekilde girilmektedir. A - Kitap Siparişleri B - Dergi Siparişleri 2- Kitap siparişlerinde kullanılan : - Para birimleri ve - Kitap geldiğinde uygulanacak işlem kodlarını görmek için PF8 tuşuna basınız.		
PF3 : Çıkış		PF8 : Sayfa Çevir

Ek 4.36. Sağlama Servisi Kitap Siparişi Bilgi Giriş Yardım Paneli-1

KU30H02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Para Birimi Kodları	Kitap Gelince Uygulanacak İşlem Kodları
USD Dolar DM Mark UKS Sterlin TL Türk Lirası HF Hollanda Florini JY Japon Yeni	01 - Kitap ve yazar adı yanlış yazılmıştır. 02 - Adres yetersizdir. 03 - Mevcudu yoktur. Sipariş iptal edilmiştir. 04 - Basılmaktadır. 05 - Yayını kesilmiştir. 06 - Kitap gelmiştir 07 - Periyodik gelmektedir. 08 - Yayınevi aboneyi kabul etmiyor. 09 - İstenilen bilgi 5 gün içinde verilmezse sipariş iptal edilir.

PF3 : Çıkış

PF7 : Önceki Sayfa

Ek 4.37. Sağlama Servisi Kitap Siparişi Bilgi Giriş Yardım Paneli-2

KU30H03

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

1- Sağlama Servisine ait siparişler 2 şekilde girilmektedir.

- 1 - Kitap Siparişleri
- 2 - Dergi Siparişleri

2- Dergi siparişlerinde kullanılan :

- Para Birimi ve
- Süreli yayın çıkış aralıklarını görmek için **PF8** tuşuna basınız.

PF3 : Çıkış

PF8 : Sayfa Çevir

Ek 4.38. Sağlama Servisi Süreli Yayın Siparişi Bilgi Giriş Yardım Paneli-1

KU30H04

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Sürelili Yayın Çıkış Aralıkları		Para Birimi Kodları
1-Yılda 1 defa	11-Yılda 11 defa	USD Dolar
2-Yılda 2 defa	12-Yılda 12 defa	DM Mark
3-Yılda 3 defa	24-Yılda 24 defa	UKS Sterlin
4-Yılda 4 defa	36-Yılda 36 defa	TL Türk Lirası
5-Yılda 5 defa	48-Yılda 48 defa	HF Hol.Florini
6-Yılda 6 defa	52-Yılda 52 defa	JY Japon Yeni
7-Yılda 7 defa	G -Günlük	
8-Yılda 8 defa	D -Düzensiz	
10-Yılda 10 defa		

PF3 : Çıkış

PF7 : Önceki Sayfa

Ek 4.39. Sağlama Servisi Sürelili Yayın Siparişı Bilgi Giriş Yardım Paneli-2

KU31M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1 - ISBN -ISSN Numarasına Göre
- 2 - Yazar Adına Göre
- 3 - Yayın Adına Göre
- 4 - Fakülte Adına Göre
- 5 - Bölüm Adına Göre
- 6- Çıkış

Tercihinizi Giriniz : 3

Enter : İşlem

Ek 4.40. Sağlama Servisi Sipariş Değişiklik-Silme İşlem Seçenekleri

KU31M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi
Yayın adı :	
PF3 : Çıkış Enter : Arama	

Ek 4.41. Sağlama Servisi Sipariş Değişiklik-Silme Bilgi Giriş Paneli

KU31M03	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	Sayfa :				
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Durum</th> <th>Sipariş Verilen Yayının Adı</th> </tr> </thead> <tbody> <tr> <td style="height: 100px;"></td> <td></td> </tr> </tbody> </table>			Durum	Sipariş Verilen Yayının Adı		
Durum	Sipariş Verilen Yayının Adı					
PF3 : Çıkış PF4 : Detayları PF7 : Geri PF8 : İleri						

Ek 4.42. Sağlama Servisi Sipariş Değişiklik-Silme Gelen Bilgi Paneli

KU31M04	ATATÜRK ÜNİVERSİTESİ		
Kitap	Merkez Kütüphanesi		
	Sağlama Servisi		
ISBN No :			
İsmarlama Tar. :	--/--/--		
Yazarı :			
Kitap Adı :			
Cilt-Seri :		Edisyon :	
Yayınlayan :			
Yayın Yeri :		Yayın Dili :	
Yayın Tarihi :		Para Birimi :	
Fiatı :			
Fak./Y.Ok./Ens. :			
Bölüm :			
Tarih - Sayı :	--/--/--	--	
Geldiği Tarih :			
Satan :			
Maliyeti :			
Gel.Yapılacak İş.:		Yer No :	
Fatura no :		Demirbaş No :	
PF3 : Çıkış		PF5 : Değişiklik	PF6 : Silme
PF7 : Birim Yazısı			

Ek 4.43. Sağlama Servisi Sipariş Değişiklik-Silme Gelen Bilgi Detay Paneli (Kitap)

KU31M05	ATATÜRK ÜNİVERSİTESİ		
Dergi	Merkez Kütüphanesi		
	Sağlama Servisi		
ISSN No :			
İsmarlama Tarihi:	--/--/--		
Dergi Adı :			
Volüm - Sayı :			
Fiziki Öz. :			
Yayın Yeri :			
Yayın Tarihi :			
Çıkış Ara. :			
Fak./Y.Ok./Ens. :			
Bölüm :			
Tarih - Sayı :	--/--/--	--	
İlk Gel. Tarih :			
Satan Firma :			
Eski Adı-1 :			
Eski Adı-2 :			
Eski Adı-3 :			
Abone Durumu :		Yayın Dili :	
PF3 : Çıkış		PF5 : Değişiklik	PF6 : Silme

Ek 4.44. Sağlama Servisi Sipariş Değişiklik-Silme Gelen Bilgi Detay Paneli (S. Yayın)

KU31H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Sağlama Servisine ait kitap isteklerinde düzeltme ve silme yaparken :

1- Ekrana önce KU31M01 paneli gelmekte ve üzerinde işlem yapılacak olan yayına ait Kitap adı, Yazar adı, fakülte veya bölüm, ISSN veya ISBN bilgileri girilip Enter tuşuna basmak suretiyle ilgili bilgilere erişim mümkün olmaktadır. Cursor yayın adının üzerinde iken PF4 tuşuna basılarak detaylara ulaşılmaktadır.

2- Gelen bilgiler üzerinde PF5 tuşu ile değişiklik , PF6 ile silme işlemi yapılabilir. PF7 tuşu ile bu isteği yapan birime, ilgili istekle alakalı yazı (Kitap Siparişleri Bölümünde) yazıcıdan alınabilmektedir.

3- Para birimi ve işlem kodları için PF8 tuşunu kullanın.

4- Durum sahasındaki 3 karakterlik bilginin anlamı ise şöyledir :

1. saha yayının cinsi ("K" ise Kitap / "D" ise Dergi)
2. saha siparişin gelip gelmediği (Sipariş gelmişse "**" , gelmemişse boşluk)
3. saha dergiler için abonelik durumu ("+" ise abonelik devam ediyor, "-" ise etmiyor)

PF3 : Çıkış

PF8 : Sayfa Çevir

Ek 4.45. Sağlama Servisi Sipariş Değişiklik-Silme Yardım Paneli

KU32M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

(Not : - Gelen Siparişleri Görmek İçin "**"
 - Gelmeyen Siparişler İçin "-"
 - Tüm Siparişleri Görmek İçin "+" giriniz.)

PF3 : Çıkış

Enter : Arama

Ek 4.46. Sağlama Servisi Sipariş Takip Paneli

KU32M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	Sayfa :				
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 70%;">Yayın Adı</th> <th style="width: 30%;">Durum</th> </tr> <tr> <td style="height: 100px;"></td> <td></td> </tr> </table>			Yayın Adı	Durum		
Yayın Adı	Durum					
PF3 : Çıkış PF4 : Ayrıntılar PF5 : Yazdır PF7 :Geri PF8 : İleri						

Ek 4.47. Sağlama Servisi Sipariş Takip Gelen Bilgi Paneli

KU32H01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri
<u>Sağlama servisine ait siparişleri izlerken :</u> 1- Siparişleri görüntülemek istiyorsanız : A- Tüm siparişleri görmek için : “*” B- Gelen siparişleri görmek için : “+” C- Gelmeyen siparişleri görmek için : “-” işaretlerini kullanarak sorgulama yapılabilir , ekrana gelen bilgiler üzerinde PF4 tuşunu kullanarak ayrıntılara göz atabilirsiniz. Ayrıntılar yayının kitap veya dergi olmasına göre farklı ekranlarda, farklı bilgiler olarak sunulmaktadır 2- Durum sahasındaki 3 karakterlik bilginin anlamı ise şöyledir : A- Saha yayın cinsi (“K” ise Kitap / “D” ise Dergi) B- Saha sipariş bilgisi (Sipariş gelmişse “*”, gelmemişse boşluk) C- Saha abonelik durumu (Abonelik devam ediyor ise “+” , etmiyorsa “-“) 3 - Ekrana gelen bilgileri PF5 tuşu ile yazıcıdan alabilirsiniz.	
PF3 : Çıkış	

Ek 4.48. Sağlama Servisi Sipariş İzleme Yardım Paneli

KU33M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1- ISSN No'dan Erişim
- 2- Yazar Adından Erişim
- 3- Yayın Adından Erişim
- 4- Yayın Yılından Erişim
- 5 -Çıkış

Tercihinizi Giriniz : 3

Enter : İşlem

Ek 4.49. Sağlama Servisi Muhasebe İşlemleri Paneli

KU33M02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

Yayın adı :

PF3 : Çıkış

Enter : Arama

Ek 4.50. Sağlama Servisi Muhasebe Bilgi Giriş Paneli

KU33M03	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	Sayfa :				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 15%;">Durum</th> <th>Sipariş Verilen Yayınlar</th> </tr> <tr> <td style="height: 150px;"></td> <td></td> </tr> </table>	Durum	Sipariş Verilen Yayınlar			
Durum	Sipariş Verilen Yayınlar					
PF3 : Çıkış PF4 : Muhasebe işlemleri PF7 : Geri PF8 : İleri						

Ek 4.51. Sağlama Servisi Muhasebe İşlemleri Gelen Bilgi Paneli

KU33M04	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	Sayfa :																
ISSN No :																		
Yayın Adı :																		
Firma Adı :																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 8%;">Yıl</th> <th style="width: 10%;">Volüm</th> <th style="width: 15%;">Fatura No</th> <th style="width: 18%;">Fatura Tarihi</th> <th style="width: 10%;">Miktarı</th> <th style="width: 10%;">Para Birimi</th> <th style="width: 10%;">Fatura Türü</th> <th style="width: 10%;">İşlem</th> </tr> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>			Yıl	Volüm	Fatura No	Fatura Tarihi	Miktarı	Para Birimi	Fatura Türü	İşlem								
Yıl	Volüm	Fatura No	Fatura Tarihi	Miktarı	Para Birimi	Fatura Türü	İşlem											
PF3 : Çıkış PF7 : Geri PF8 : İleri PF9 : Yazdır Enter : İşlem																		

Ek 4.52. Sağlama Servisi Muhasebe İşlemleri Bilgi Giriş Paneli

KU33H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Sağlama Servisine ait muhasebe bilgileri girişi yapılırken :

1- Ekrana önce ISSN No, yayın adı, yayın yılı, yazar adından tarama yapılırken seçenekler ekrana gelmekte, kullanıcı istediği bilgilere bu seçenekler yolu ile erişebilmektedir. Örneğin ;

- 1- **Para%** : Adı para ile başlayan yayınlara erişim
- 2- **%Para** : Adının son kelimeleri para ile biten yayınlara erişim
- 3- **%Para%** : Adının içinde para kelimesi geçen yayınlara erişim

Bu örnekler yazar, ISSN No için de aynen geçerlidir. Yıl aynen girilmelidir.

2- Ekrana gelen yayınların içersinde siparişi gelen bilgiler için siparişin gelip gelmediği ayrıca abonelik durumunun devam edip etmediği ve yayının cinsi görülmektedir.

3- **PF4** tuşu ile yayınların muhasebe işlemlerine geçilmekte, yayın kitap ise sadece sipariş bilgisi görüntülenmektedir. İşlem hanesine kayıt için **K**, değişiklik için **D** ve silme için **S** harfi kullanılmalıdır. **PF7 -PF8** tuşları ile ekrandaki bilgiler arasında hareket edilebilmektedir. Para birim kodları için **PF8** tuşuna basınız.

4- **PF9** tuşu ile muhasebe bilgileri yazıcıdan alınabilmektedir.

F3 : Çıkış

PF8 : Sayfa Çevir

Ek 4.53. Sağlama Servisi Muhasebe İşlemleri Yardım Paneli-1

KU34M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1 - Sale Invoice Fatura No
- 2 - Extra Charge Fatura No
- 3 - Credit Note Fatura No
- 4 - Çıkış

Tercihinizi Giriniz : 1

Enter : İşlem

Ek 4.54. Sağlama Servisi Faturaların Takip İşlemleri Paneli

KU34M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi
Sale Invoice No :	
PF3 : Çıkış Enter : Arama	

Ek 4.55. Sağlama Servisi Faturaların Takibi Bilgi Giriş Paneli

KU34M03	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	Sayfa :										
Firma Adı :												
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">ISSN No</th> <th style="width: 40%;">Yayın Adı</th> <th style="width: 15%;">Volüm</th> <th style="width: 20%;">Fatura Tutarı</th> <th style="width: 10%;">Para Birimi</th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>			ISSN No	Yayın Adı	Volüm	Fatura Tutarı	Para Birimi					
ISSN No	Yayın Adı	Volüm	Fatura Tutarı	Para Birimi								
PF3 : Çıkış PF7 : Geri PF8 : İleri PF9 : Yazdır												

Ek 4.56. Sağlama Servisi Faturaların Takibi Gelen Bilgi Paneli

KU34H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Sağlama Servisie ait Faturaları görmek için :

1- Faturalar 3 türde izlenebilmektedir.

- A- Sale Invoice
- B- Extra Change
- C- Credit Note

2- Fatura türünü seçtikten sonra, aşağıdaki gibi numarasını girip **Enter** tuşuna basılarak istenilen bilgiye ulaşılmaktadır. **Örnek :**

- 1- 123% : 123 ile başlayan fatura numaralarını görmek için
- 2- %123 : 123 ile biten fatura numaralarını görmek için
- 3- %123% : içerisinde 123 rakamı geçen faturaları görmek için

3- PF9 tuşu ile ekrana gelen bilgiler yazıcıdan alınabilmektedir.

PF3 : Çıkış

Ek 4.57. Sağlama Servisi Faturaların Takibi Yardım Paneli-1

KU35M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1 - Kitaplar Parasal Dağılım
- 2 - Dergiler Parasal Dağılım
- 3- Çıkış

Tercihiniz Giriniz :

Tarama Başlangıç Yılı :

Tarama Bitiş Yılı :

Enter : İşlem

Ek 4.58. Sağlama Servisi Faturaların Dağılımı Bilgi Giriş Paneli

KU35M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	
--- / --- yılları arası faturaların parasal dağılımı		
Para Birimi		
PF3 : Çıkış		PF9 : Yazdır

Ek 4.59. Sağlama Servisi Faturaların Dağılımı Gelen Bilgi Paneli

KU36M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Sağlama Servisi	
Firma Adı	Toplam	
PF3 : Çıkış		

Ek 4.60. Sağlama Servisi Faturaların Firma Bazında Dağılımı Bilgi Paneli

KU37M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sağlama Servisi

- 1 - Yayın Adına Göre
2 - Firma Adına Göre
3- Çıkış

Tercihiniz Giriniz :

PF3 : Çıkış

Ek 4.61. Sağlama Servisi Süreli Yayın Listesi Alma Paneli

KU40M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Ödünç Verme Servisi

Kart Numarası :
Üyelik Tarihi : --/--/---
Sicil / Öğr. No :
Adı :
Soyadı :
Bölümü :
Ünvanı :
Telefon :
Üyelik Durumu :

PF3 : Çıkış PF4 : Kayıt PF5 : Değişiklik PF6 : Silme PF9 : Ekran Silme Enter : Arama

Ek 4.62. Ödünç Verme Servisi Okuyucu Kayıt İşlemleri Paneli

KU40H01

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Yardım Bilgileri

Üye Kimlik Bilgilerini girerken şu hususlara dikkat edilmelidir :

1- Üyelik işlemlerinin çalışma esasını **Kart no** ve **Sicil no** oluşturmakta, dolayısı ile işlemlerde bu iki bilginin doğruluğu önem kazanmaktadır.

Kart No : Üyelerden herbirine bilgisayar tarafından verilen numara

Sicil No :

A - Eğer şahıs öğrenci ise 9 haneli okul numarası

B - Diğer şahıslar için kurum sicil numarası olarak girilmelidir.

2- Daha önceden kaydı yapılmış bir şahsın bilgilerine **Kart** ve **Sicil No** girildikten sonra **Enter** tuşuna basmak suretiyle erişilebilir. **PF5** tuşu ile gelen bilgiler üzerinde değişiklik , **PF6** tuşu ile silme işlemleri yapılabilir.

3- Üye bilgi kayıt programında dikkat edilecek diğer husus bilgisi girilecek kişiye ait ünvan kodudur. **PF8** tuşu ile bu kodlara bakabilirsiniz . Bu bilgi çeşitli raporlar, istatistikler tutmak için konulmuştur.

PF3 : Çıkış

PF8 : Ünvan Kodları

Ek 4.63. Ödünç Verme Servisi Okuyucu Kayıt İşlemleri Yardım Paneli-1

KU40H02

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Yardım Bilgileri

Ünvan Kodları	Ünvan Kodları	Ünvan Kodları
1 - Profesör 2 - Yrd. Doç 3 - Doçent 4 - Doktor 5 - Doktora Öğrencisi 6 - Arş. Gör. 7 - Lisans Öğrencisi 8 - Yüksek Lisans Öğ. 9 - Memur U - Uzman O - Okutman		

PF3 : Çıkış

PF7 : Önceki Sayfa

Ek 4.64. Ödünç Verme Servisi Okuyucu Kayıt İşlemleri Yardım Paneli-2

KU41M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ödünç Verme Servisi
Adı : Soyadı :	
PF3 : Çıkış	Enter : Arama

Ek 4.65. Ödünç Verme Servisi Okuyucu Arama Bilgi Giriş Paneli

KU41M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ödünç Verme Servisi	Sayfa :						
Yayın, okuyucuya verilecek ise : O Okuyucu, yayını teslim edecekse : G giriniz.								
Okuyucunun :	<table border="1"> <thead> <tr> <th>Adı</th> <th>Soyadı</th> <th>Kart Numarası</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Adı	Soyadı	Kart Numarası				
Adı	Soyadı	Kart Numarası						
PF3 : Çıkış	PF4 : İşlemler	PF7 : Geri						
PF8 : İleri								

Ek 4.66. Ödünç Verme Servisi Ödünç Verme/Geri Alma İşlemleri Bilgi Giriş Paneli

KU42M01	ATATÜRK ÜNİVERSİTESİ		
	Merkez Kütüphanesi		
	Ödünç Verme Servisi		
Kart No :		Ünvan :	
Sicil No :		Kayıt T. :--/------	
Adı :		Telefon :	
Soyadı :			
Bölümü :			

Ödünç verilecek yayının adı	Yıl	Cilt	Verilecek olan sayılar

PF3 : Çıkış
PF4 : Kayıt

Ek 4.67. Ödünç Verme Servisi Ödünç Verme İşlemleri Bilgi Giriş Paneli

KU42H01	ATATÜRK ÜNİVERSİTESİ
	Merkez Kütüphanesi
	Yardım Bilgileri

Yayınları ödünç verirken şu sıra takip edilmelidir :

- 1-**Kart no** veya **Sicil no** hanelerine kitabı vereceğiniz şahsın bilgilerini girip **ENTER** tuşuna basarak o şahsın kayıtlı olup olmadığını öğreniniz.
- 2- Herhangi bir hata oluşmamış ise o şahsın bilgileri ekrana gelir ve kursör **Kitap-Dergi adı** sahasına yerleşir.
- 3- Bu aşamada verilecek kitap-dergi adı ilgili bölüme yazıldıktan sonra :
 - A- Verilecek belge **Kitap** ise **cilt**, **seri** kodlanmayacak
 - B- Verilecek belge **Sürelî yayın** ise **cilt**, **seri** kodlanacaktır
- 4- Tekrar eden kitap adlarını yazmamak için “=” kullanabilir.
- 5- Son olarak da kitabın alındığı tarih **gün-ay-yıl** olarak kodlanıp **PF4** tuşu kayıt yapılacaktır.

PF3 : Çıkış

Ek 4.68. Ödünç Verme Servisi Ödünç Verme İşlemleri Yardım Paneli

KU43M01		ATATÜRK ÜNİVERSİTESİ		
		Merkez Kütüphanesi		
		Ödünç Verme Servisi		
Kart No	:	Ünvan	:	
Sicil No	:	Kayıt T.	:-/--/----	
Adı	:	Telefon	:	
Soyadı	:			
Bölümü	:			

Verilen yayının adı	Alış tarihi	Yıl	Cilt	Verilmiş olan sayılar

PF3 : Çıkış PF4 : İşlem PF5 : Yazdır PF7 : Geri PF8 : İleri

Ek 4.69. Ödünç Verme Servisi Ödünç Verilen Yayın Bilgileri Paneli

KU43H01	ATATÜRK ÜNİVERSİTESİ
	Merkez Kütüphanesi
	Yardım Bilgileri

Ödünç verilen yayın geri alınırken şu sıra takip edilmelidir :

1- **Kart no** veya **Sicil no** hanelerine kitabı vereceğiniz şahsın bilgilerini girip **ENTER** tuşuna basarak o şahsın aldığı ve iade etmediği bütün kitap ve dergilere ulaşabilirsiniz.

2- Herhangi bir hata oluşmamış ise o şahsın bilgileri ve alıp da iade etmediği tüm kitap ve dergiler **Kitap - Dergi adı** sahasına yerleşir. **PF8** veya **PF7** tuşları kullanılarak ekrandaki bilgiler üzerinde ileri geri hareket edebilirsiniz.

3- Bu aşamada iade edilecek kitap adının bulunduğu satıra iade tarihi girilip **PF4** tuşu ile bilgileri kaydediniz .

4- Tekrar eden iade tarihleri için “=” kullanabilir.

5- **PF3** tuşu ile çıkıp diğer iadeler için bu aşamalar tekrarlanmalıdır.

PF3 : Çıkış

Ek 4.70. Ödünç Verme Servisi Geri Alma İşlemleri Yardım Paneli

KU44M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ödünç Verme Servisi
Yayını Alan :	
Geçen Süre :--/--/---	
Bölümü :	
Telefon :	

Kart No	Yayın Adı

PF3 : Çıkış
PF4 : İşlem
PF5 : Yazdır
PF7 : Geri
PF8 : İleri

Ek 4.71. Ödünç Verme Servisi Süresi Geçen Şahısları Belirleme Paneli

KU44H01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri
<u>Süresi dolan kitap ve dergilere ait bilgiler :</u>	
1- Ana panelden süresi dolan (Kitaplarda 30 gün, dergilerde 15 gün) belgeler bölümü çalıştırılınca ekrana o kitabı alan şahsın kart numarası ve kitap adı bilgileri görüntülenir. PF8 tuşu ile ileri, PF7 tuşu ile geriye hareket edilebilir. 2- Kart No sahasına cursor hareket ettirilerek ilgili sahada PF4 tuşuna basarak o kitabı alan şahsın adı ve yayını aldığından bu yana geçen süre yıl-ay-gün olarak ekranda görüntülenebilir.	
PF3 : Çıkış	

Ek 4.72. Ödünç Verme Servisi Süresi Geçen Şahısları Yardım Paneli

KU45M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ödünç Verme Servisi	Sayfa :
<u>Aranan Yayının :</u> Adı : Yıl : Cilt : Sayı :		
PF3 : Çıkış		Enter : Arama

Ek 4.73. Ödünç Verme Servisi Herhangi Bir Yayın Arama Paneli

KU45M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ödünç Verme Servisi	Sayfa :										
Adı Soyadı : Ünvanı : Telefon : Bölümü :												
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Verilen yayının adı</th> <th style="width: 15%;">Yayın Yılı</th> <th style="width: 10%;">Cilt</th> <th style="width: 30%;">Verilen sayılar</th> <th style="width: 15%;">Kart No</th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>			Verilen yayının adı	Yayın Yılı	Cilt	Verilen sayılar	Kart No					
Verilen yayının adı	Yayın Yılı	Cilt	Verilen sayılar	Kart No								
PF3 : Çıkış	PF7 : Geri	PF8 : İleri										
PF9 : Ara												

Ek 4.74. Ödünç Verme Servisi Aranan Yayın Gelen Bilgi Paneli

KU45H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Herhangi bir kitap veya süreli yayın kimde olduğunu görmek için :

1- Aranan bilgi girilip **Enter** tuşuna basılarak verilmiş olan yayına ait bilgilere erişebilir. Yayın adı tam hatırlanmıyorsa , Örneğin “chemistry” tam yazılmak isteniyorsa %chem% şeklinde yazıp, **Enter** tuşuna basın.

2- **Konu No** veya yayına ait yıl, cilt, sayı bilgileri de girilip arama yapılabilir.

3- Gelen bilgiler üzerinde kursor **Kart No** sahasına yerleşir. İstenilen **Kart No** üzerine gelip **PF9** tuşu ile o yayının kimde olduğu görülebilir.

PF3 : Çıkış

Ek 4.75. Ödünç Verme Servisi Aranan Yayın Yardım Paneli

KU46M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Ödünç Verme Servisi

Ünvanı	Alış Sayısı

PF3 : Çıkış

Ek 4.76. Ödünç Verme Servisi Okuyucu Grupları Dağılım Paneli

KU47M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Ödünç Verme Servisi

Aylık - Yıllık İşlem Akışı

İlgili yıl / ay bilgisini giriniz :

YILI **AYI**
 — —

Yapılan işlem sayısı	: —
Alınup iade edilen yayın sayısı	: —
Alınup iade edilmeyen yayın sayısı	: —

PF3 : Çıkış
Enter : İşlem

Ek 4.77. Ödünç Verme Servisi Yapılan İşlem Sayıları Paneli

KU51M01

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Ölçek : 1 / 25.000

Kayıtlı harita adedi :

Sayfa :

Harita yer adı	A	B	C	D	Toplam	İşlem

PF3 :Çıkış

Enter : İşlem

PF7 : Geri

PF8 : İleri

PF9 : Yazdır

Ek 4.78. Dökümantasyon Servisi 1/25000 Ölçekli Haritalar Bilgi Giriş Paneli

KU51H01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Harita bilgilerini girerken şu husulara dikkat edilmelidir :

1- Daha önceden kaydetmiş olduğunuz tüm 1/25.000 ölçekli harita bilgileri ekranda görüntülenmekte ve işlem yapılabilir. Şu anda kullandığınız limit 300 kayıt ile sınırlıdır. Bu demektir ki bir anda 300 kayda erişebilirsiniz

2- Görüntülenen bu kayıtlar üzerinde silme , değişiklik veya yeni kayıt bilgileri girilebilmektedir. İşlem hanesine :

- Kayıt yapacaksanız "K"
- Değişiklik için "D"
- Silme yapılırsa "S" harflerini girip Enter tuşuna basınız.

3- PF9 tuşuna basılarak ilgili harita bilgileri yazıcıdan alınabilmektedir.

PF3 : Çıkış

Ek 4.79. Dökümantasyon Servisi 1/25000 Ölçekli Haritalar Yardım Paneli

KU52M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Ölçek : 1 / 100.000

Kayıtlı harita adedi :

Sayfa :

İlin adı	Pafta no	İlin adı	Pafta no	Toplam	İşlem

PF3 : Çıkış

Enter : İşlem

PF7 : Geri

PF8 : İleri

PF9 : Yazdır

Ek 4.80. Dökümantasyon Servisi 1/100000 Ölçekli Haritalar Bilgi Giriş Paneli

KU53M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Ölçek : 1 / 250.000								
Kayıtlı harita adedi :	Sayfa :								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 25%;">İlin adı</th> <th style="width: 25%;">Plaka no</th> <th style="width: 25%;">Toplam</th> <th style="width: 25%;">İşlem</th> </tr> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> </tr> </table>		İlin adı	Plaka no	Toplam	İşlem				
İlin adı	Plaka no	Toplam	İşlem						
PF3 : Çıkış Enter : İşlem PF7 : Geri PF8 : İleri PF9 : Yazdır									

Ek 4.81. Dökümantasyon Servisi 1/250000 Ölçekli Haritalar Bilgi Giriş Paneli

KU54M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Hava Fotoğrafları										
Kayıtlı harita adedi :	Sayfa :										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 15%;">Ebadi</th> <th style="width: 15%;">Özel no</th> <th style="width: 40%;">Notlar ve açıklamalar</th> <th style="width: 15%;">Adet</th> <th style="width: 15%;">İşlem</th> </tr> <tr> <td style="height: 150px;"></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>		Ebadi	Özel no	Notlar ve açıklamalar	Adet	İşlem					
Ebadi	Özel no	Notlar ve açıklamalar	Adet	İşlem							
PF3 : Çıkış Enter : İşlem PF7 : Geri PF8 : İleri PF9 : Yazdır											

Ek 4.82. Dökümantasyon Servisi Hava Fotoğrafları Bilgi Giriş Paneli

TA10M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yayın Tarama

- 1 - Kitap Tarama
- 2 - Süreli Yayın Tarama
- 3 - Tez ve Makale Tarama
- 4 - Harita Bilgileri Tarama
- 5 - Çıkış

Tercihinizi Giriniz :__

Enter : İşlem

Ek 4.83. Tarama İşlemleri Ana Menü

TA20M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Kitap Tarama

- 1 - Kitap Adına Göre
- 2 - Yazar Adına Göre
- 3 - Konu Numarasına Göre
- 4 - Konu Adına Göre
- 5 - Seri Adına Göre
- 6 - Diğer Emeği Geçenlere Göre *
- 7 - ISBN Numarasına Göre
- 8 - Yayın Yılına Göre
- 9 - Yayın Diline Göre
- 10- Çıkış

Tercihinizi Giriniz :__

Enter : İşlem

Ek 4.84. Kitap Tarama Seçenekleri Paneli

TA30M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayın Tarama

- 1 - Yayın Adına Göre
- 2 - Yayın Yılına Göre
- 3 - Yayınlayana Göre
- 4 - ISSN Numarasına Göre
- 5 - Çıkış

Tercihinizi Giriniz :__

Enter : İşlem

Ek 4.85. Sürelî Yayın Tarama Seçenekleri Paneli

TA40M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Tez ve Makale Tarama

- 1 - Tez-Makale Adına Göre
- 2 - Tez-Makale Yazarına Göre
- 3 - Tez-Makale Konu Numarasına Göre
- 4 - Tez-Makale Konu Adına Göre
- 5 - Tez-Makale Yayınlayan Fakülte-Bölüme Göre
- 6 - Tez-Makale Yayın Yılına Göre
- 7 - Tez-Makale Yayın Diline Göre
- 8 - Çıkış

Tercihinizi Giriniz :__

Enter : İşlem

Ek 4.86. Tez-Makale Tarama Seçenekleri Paneli

TA50M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Haritalar
1 - 1 / 25.000 Ölçekli 2 - 1 / 100.000 Ölçekli 3 - 1 / 250.000 Ölçekli 4 - Hava Fotoğrafları 5 - Çıkış	
Tercihinizi Giriniz :__	
Enter : İşlem	

Ek 4.87. Harita Tarama Seçenekleri Paneli

TA21M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kitap Tarama				
Kitap adını giriniz :					
1. bilgi 2. bilgi	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 80%; text-align: center; padding: 5px;">İşlem</td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center; padding: 5px;">-----</td> <td></td> </tr> </table>	İşlem		-----	
İşlem					

<u>NOT :</u> Bu programı ilk defa kullanıyorsanız PF1 tuşuna basınız.					
PF3 : Çıkış Enter : Arama					

Ek 4.88. Kitap Tarama Bilgi Giriş Paneli

TA21M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Kitap Tarama	Sayfa :
Aranan :		
Taraması yapılan bilgi ile alakalı olarak	adet yayın bulunmuştur.	
Yayın adı :		

Ek 4.89. Kitap Tarama Gelen Bilgi Paneli

TA21H01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri
<u>Açıklamalar :</u> 1- Taramak istediğiniz bölüm “Kitap Adından”ve aradığınız bilgi de “Program” olsun. İlk bilgi alanına veri şu şekilde girilmelidir. A- Program % : Program kelimesi ile başlayan kayıtları seçer. B- %Program : Program kelimesi ile biten kayıtları seçer. C- %Program% : İçerisinde Program kelimesi geçen kayıtları seçer. 2- İşlem sahasına sadece VE-VEYA kelimeleri girilebilmektedir ve şu durumlarda kullanılmalıdır. A- İçerisinde hem bilgisayar hemde program kelimesinin ikisinin birden birden geçtiği yayınları taramak için bilgi hanesine bilgisayar, işlem hanesine VE, 2.bilgi hanesine de program kelimesi girilmelidir. B- İçerisinde bilgisayar ve program kelimelerinden herhangi birisinin geçtiği yayınları tarama için ilk bilgi hanesine bilgisayar işlem hanesine VEYA, 2.bilgi hanesine de program kelimesi girilmelidir. C - Bu saha Yayın dili ve ISBN No taramalarında kullanılmaz.	
PF3 : Çıkış	PF8 : Sayfa Çevir

Ek 4.90. Kitap Tarama Yardım Paneli-1

TA21H02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

2- Yayın Yılına göre tarama yapılacak ise bilgi aşağıdaki gibi girilmelidir.

- A- “= 1990” (YayınYılı 1990 olan yayınlara erişim)
 B- “=>1990” (YayınYılı 1990 ve 1990 dan büyük olan yayınlara erişim)
 C- “> 1990 ” (YayınYılı 1990 dan büyük yayınlara erişim)
 D- “<=1990” (YayınYılı 1990 ve 1990 dan küçük olan yayınlara erişim)
 E- “= 1990 ” (YayınYılı 1990 dan küçük olan yayınlara erişim)

3- Yayın Dili taramalarında aşağıdaki dil kodlarını büyük harf olarak giriniz. **Örneğin**, Yayın Dili İngilizce olan yayınları taramak için “%ING%” şeklinde giriniz.

4- Dil Kodları :

TUR	: Türkçe	ING	: İngilizce
OSM	: Osmanlıca	FRA	: Fransızca
ARP	: Arapça	ITA	: İtalyanca
RUS	: Rusça	ALM	: Almanca

PF3 : Çıkış

PF7 : Önceki Sayfa

Ek 4.91. Kitap Tarama Yardım Paneli-2

TA21H03

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Yayın tarama bölümüne ilişkin notlar :

1- Yayın tarama bölümünde -**hangi seçenikle tararsanız tarayın**- ekrana önce tarama yaptığınız yayınların isimleri gelmektedir. 10'dan fazla yayın bulunduğu zaman **PF8** tuşu ile ileri **PF7** tuşu ile geriye doğru hareket edebilirsiniz. Programın şu anki yayın tarama limiti 500 olarak tesbit edilmiştir.

2- **PF5** tuşu ile, elde edilen sonuçların tümüne ait yazıcı dökümleri alınabilmektedir. Bu bölümü gereksiz yere kullanmayınız.

3- Cursor, hakkında ayrıntılı bilgi almak istenilen yayının üzerinde iken **PF4** tuşuna basılır ise “kitaplara ait katalog kartı” ekrana gelecektir.

PF3 : Çıkış

Ek 4.92. Kitap Tarama Yardım Paneli-3

TA21H04

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Bu ekranın en üstünde iki tür bilgi yer almaktadır :

1- Yayının şu anda nerede bulunduğu (Kütüphanede-Dışarıda-Depoda-Belirsiz)

2- Yayının okuyuculara verilip verilmeyeceği. Bundan sonra gelen 3 satırda ilk önce yayın türü (Yayın normal kitap ise herhangi bir bilgi yok, fakat **Müracaat Eseri, Nadir Eser, El Yazması** ise yayın türü kodu ile belirtilir). Yazar ve yazara ait ayırıcı kod bulunmakta, hemen ardındaki satırda Yazar Adı bilgisi yer almaktadır. Yazar adından sonraki bölümler paragraflar şeklinde ayrılmıştır.

Birinci paragraf : Kitap adı, yazar adı ve diğer emeği geçenler ile baskı kaydı, yayın yeri, yayın yılı, yayınlayan, yer almaktadır.

İkinci paragraf : Yayına ait sayfa-cilt bilgisi, fiziki özellikleri, varsa seri adı bulunmaktadır.

3. ve diğerleri : Paragraflar halinde bibliyografya, açıklayıcı ek bilgiler, dizin bilgisi, ISBN No bilgileri vardır.

Son paragraf : Katalog kartlarına ait izleme bilgilerini verir.

PF3 : Çıkış

Ek 4.93. Kitap Tarama Yardım Paneli-4

TA31M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Sürelî Yayın Tarama

Sürelî yayın adını giriniz :		İşlem
1. bilgi		-----
2. bilgi		

NOT : Bu programı ilk defa kullanıyorsanız PF1 tuşuna basınız.

PF3 : Çıkış

Enter : Arama

Ek 4.94. Sürelî Yayın Tarama Bilgi Giriş Paneli

TA31M02

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Sürelî Yayın Tarama

Sayfa :

Taraması yapılan bilgi ile alakalı olarak adet yayın bulunmuştur.

Yayın adı :

PF2 : Çıkış

PF4 : Ayrıntılı Bilgi

PF5 : Yazıcı

PF7 : Geri

PF8 : İleri

Ek 4.95. Sürelî Yayın Tarama Gelen Bilgi Paneli

TA31H01

ATATÜRK ÜNİVERSİTESİ

Merkez Kütüphanesi

Yardım Bilgileri

Açıklamalar :

1 - Taramak istediğiniz bölüm “Sürelî Yayın Adı” ve aradığınız bilgi de “**Program**” olsun. İlk bilgi alanına veri şu şekilde girilmelidir.

A- Program % : Program kelimesi ile başlayan kayıtları seçer.

B- %Program : Program kelimesi ile biten kayıtları seçer.

C- %Program% : İçerisinde Program kelimesi geçen kayıtları seçer.

2- İşlem sahasına sadece **VE-VEYA** kelimeleri girilebilmektedir ve şu durumlarda kullanılmalıdır.

A- İçerisinde hem bilgisayar hemde program kelimesinin ikisinin birden birden geçtiği yayınları taramak için bilgi hanesine bilgisayar, işlem hanesine **VE**, 2.bilgi hanesine de program kelimesi girilmelidir.

B- İçerisinde bilgisayar ve program kelimelerinden herhangi birisinin geçtiği yayınları tarama için ilk bilgi hanesine bilgisayar, işlem hanesine **VEYA**, 2.bilgi hanesine de program kelimesi girilmelidir. Bu saha **Yayın dili** ve **ISBN No** taramalarında kullanılmaz.

PF3 : Çıkış

PF8 : Sayfa Çevir

Ek 4.96. Sürelî Yayın Tarama Yardım Paneli-1

TA31H02

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

3- Yayın Yılına göre tarama yapılacaksa 1. bilgi alanına şu şekilde girilmelidir.

- A- “>= 1990” (1990 yılına eşit ve büyük olan Yayın Yıllarını tarar.)
- B- “<= 1990” (1990 yılına eşit ve küçük olan Yayın Yıllarını tarar.)
- C- “= 1990” (1990 yılına eşit olan Yayın Yıllarını tarar.)
- D- “< 1990” (1990 yılından küçük olan Yayın Yıllarını tarar.)
- E- “> 1990” (1990 yılından büyük olan Yayın Yıllarını tarar.)

4- Sadece “=”, “>”, “<” kullanılacaksa yukarıdaki örnekteki gibi boşluk bırakınız

PF3 : Çıkış

PF7 : Önceki Sayfa

Ek 4.97. Süreli Yayın Tarama Yardım Paneli-2

TA31H03

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Süreli Yayın Tarama bölümüne ilişkin notlar :

1- Yayın tarama bölümünde -hangi seçenikle tararsanız tarayın- ekrana önce tarama yaptığınız yayınların isimleri gelmektedir. 10 ' dan fazla yayın bulunduğu zaman PF8 tuşu ile ileri, PF7 tuşu ile geriye doğru hareket edebilirsiniz. Programın şu anki yayın tarama limiti 500 olarak tesbit edilmiştir.

2- PF5 tuşu ile, elde edilen sonuçların tümüne ait yazıcı dökümleri alınabilmektedir. Bu bölümü gereksiz yere kullanmayınız.

3- Cursor, hakkında ayrıntılı bilgi almak istenilen yayının üzerinde iken PF4 tuşuna basılır ise “süreli yayınlara ait katalog kartı” ekrana gelecektir.

PF3 : Çıkış

Ek 4.98. Süreli Yayın Tarama Yardım Paneli-3

TA31H04

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Yardım Bilgileri

Bu ekranda süreli yayınlar katalog kartı bilgileri bulunmaktadır.

Birinci Paragraf : Süreli yayın adı, cilt-volüm-sayı-yıl, yayın yeri, yayınlayan, aboneliğin devam edip etmediği v.b. bilgileri yer almaktadır.

İlk paragrafın sonunda Yıl bilgisinin sonunda :

- a) “.” varsa abonelik kesilmiş
- b) “-“ varsa abonelik hala devam ediyor anlamına gelmektedir.

* Yayın yerinden önceki bilgi

- a) “- —“ ise yayının çıkmakta olduğunu
- b) “.” ise yayının kesildiğini göstermektedir.

İkinci Paragraf : Yayına ait sayfa-cilt bilgisi, fiziki özellikleri v.b. tanıtıcı bilgiler bulunmaktadır.

3.ve Diğerleri : Paragraf halinde çıkış periyodu, ISSN No, yayına ait varsa eski isimleri, açıklayıcı bilgiler vardır.

PF3 : Çıkış

Ek 4.99. Süreli Yayın Tarama Yardım Paneli-4

TA41M01

ATATÜRK ÜNİVERSİTESİ
Merkez Kütüphanesi
Tez ve Makale Tarama

	Tez ve makale adını giriniz :	İşlem
1. bilgi		
2. bilgi		

NOT : Bu programı ilk defa kullanıyorsanız PF1 tuşuna basınız.

PF3 : Çıkış

Enter : Arama

Ek 4.100. Tez-MakaleYayın Tarama Bilgi Giriş Paneli

TA41M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Tez ve Makale Arama	Sayfa :				
Aranan :						
Taraması yapılan bilgi ile alakalı olarak adet yayın bulunmuştur.						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 80%;">Yayın adı</th> <th style="width: 20%;">Türü</th> </tr> <tr> <td style="height: 100px;"></td> <td></td> </tr> </table>		Yayın adı	Türü			
Yayın adı	Türü					
PF2 : Çıkış PF4 : Ayrıntılı Bilgi PF5 : Yazıcı PF7 : Geri PF8 : İleri						

Ek 4.101. Tez-MakaleYayın Tarama Gelen Bilgi Paneli

TA41H01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri	
<u>Açıklamalar :</u>		
1 - Taramak istediğiniz bölüm “ Tez Adından Tarama ” ve aradığınız bilgi de “ Program ” olsun. İlk bilgi alanına veri şu şekilde girilmelidir.		
A- Program % : Program kelimesi ile başlayan kayıtları seçer. B- %Program : Program kelimesi ile biten kayıtları seçer. C- %Program% : İçerisinde Program kelimesi geçen kayıtları seçer.		
2- İşlem sahasına sadece VE-VEYA kelimeleri girilebilmektedir ve şu durumlarda kullanılmalıdır.		
A- İçerisinde hem bilgisayar hemde program kelimesinin ikisinin birden birden geçtiği yayınları taramak için bilgi hanesine bilgisayar, işlem hanesine VE, 2.bilgi hanesine de program kelimesi girilmelidir.		
B- İçerisinde bilgisayar ve program kelimelerinden herhangi birisinin geçtiği yayınları tarama için ilk bilgi hanesine bilgisayar, işlem hanesine VEYA, 2.bilgi hanesine de program kelimesi girilmelidir. Bu saha Yayın dili ve ISBN No taramalarında kullanılmaz.		
PF3 : Çıkış PF8 : Sayfa Çevir		

Ek 4.102. Tez-MakaleYayın Tarama Yardım Paneli-1

TA41H02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri								
2- Yayın Yılına göre tarama yapılacak ise bilgi aşağıdaki gibi girilmelidir. A- “= 1990” (YayınYılı 1990 olan yayınlara erişim) B- “=>1990” (YayınYılı 1990 ve 1990 dan büyük olan yayınlara erişim) C- “> 1990 ” (YayınYılı 1990 dan büyük yayınlara erişim) D- “<=1990” (YayınYılı 1990 ve 1990 dan küçük olan yayınlara erişim) E- “= 1990 ” (YayınYılı 1990 dan küçük olan yayınlara erişim) 3- Yayın Dili taramalarında aşağıdaki dil kodlarını büyük harf olarak giriniz. Örneğin, Yayın Dili İngilizce olan yayınları taramak için “%ING%” şeklinde giriniz. 4- Dil Kodları : <table> <tr> <td>TUR : Türkçe</td> <td>ING : İngilizce</td> </tr> <tr> <td>OSM : Osmanlıca</td> <td>FRA : Fransızca</td> </tr> <tr> <td>ARP : Arapça</td> <td>ITA : İtalyanca</td> </tr> <tr> <td>RUS : Rusça</td> <td>ALM : Almanca</td> </tr> </table>		TUR : Türkçe	ING : İngilizce	OSM : Osmanlıca	FRA : Fransızca	ARP : Arapça	ITA : İtalyanca	RUS : Rusça	ALM : Almanca
TUR : Türkçe	ING : İngilizce								
OSM : Osmanlıca	FRA : Fransızca								
ARP : Arapça	ITA : İtalyanca								
RUS : Rusça	ALM : Almanca								
PF3 : Çıkış	PF7 : Önceki Sayfa								

Ek 4.103. Tez-MakaleYayın Tarama Yardım Paneli-2

TA41H03	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri
<u>Tez Tarama bölümüne ilişkin notlar :</u> 1- Yayın tarama bölümünde -hangi seçenikle tararsanız tarayın- ekrana önce tarama yaptığınız yayınların isimleri gelmektedir. 10’ dan fazla yayın bulunduğu zaman PF8 tuşu ile ileri, PF7 tuşu ile geriye doğru hareket edebilirsiniz. Programın şu anki yayın tarama limiti 500 olarak tesbit edilmiştir. 2- PF5 tuşu ile, elde edilen sonuçların tümüne ait yazıcı dökümleri alınabilmektedir. Bu bölümü gereksiz yere kullanmayınız. 3- Cursor, hakkında ayrıntılı bilgi almak istenilen yayının üzerinde iken PF4 tuşuna basılır ise “tezlere ait katalog kartı” ekrana gelecektir.	
PF3 : Çıkış	

Ek 4.104. Tez-MakaleYayın Tarama Yardım Paneli-3

TA41H04	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Yardım Bilgileri
Bu ekranın en üstündeki bilgi Tez Türünü göstermektedir. Bundan sonra gelen 2 satırda ilk önce konu numarası, altında ise yazar kodu bulunmaktadır.Alt satırda yazar adı bilgisi yer almaktadır. * Yazar adından sonraki bölümler paragraflar şeklinde ayrılmıştır. Birinci paragraf : Tez, Makale adı, yazarı ve diğer emeği geçenler ile baskı kaydı, yayın yeri, yayınlayan ve yayın yılı yer almaktadır. İkinci paragraf : Yayına ait sayfa-cilt bilgisi, fiziki özellikleri, varsa Seri Adı bulunmaktadır. 3. ve diğerleri : Paragraflar halinde bibliyografya, açıklayıcı ek bilgiler, dizin bilgisi v.b. bilgiler vardır.	
PF3 : Çıkış	

Ek 4.105. Tez-MakaleYayın Tarama Yardım Paneli-4

TA51M01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Harita Bilgileri
Harita adını giriniz :	
Not : Bu programı ilk defa kullanıyorsanız PF1 tuşuna basınız.	
PF3 : Çıkış	Enter : Arama

Ek 4.106. Harita Tarama Bilgi Giriş Paneli

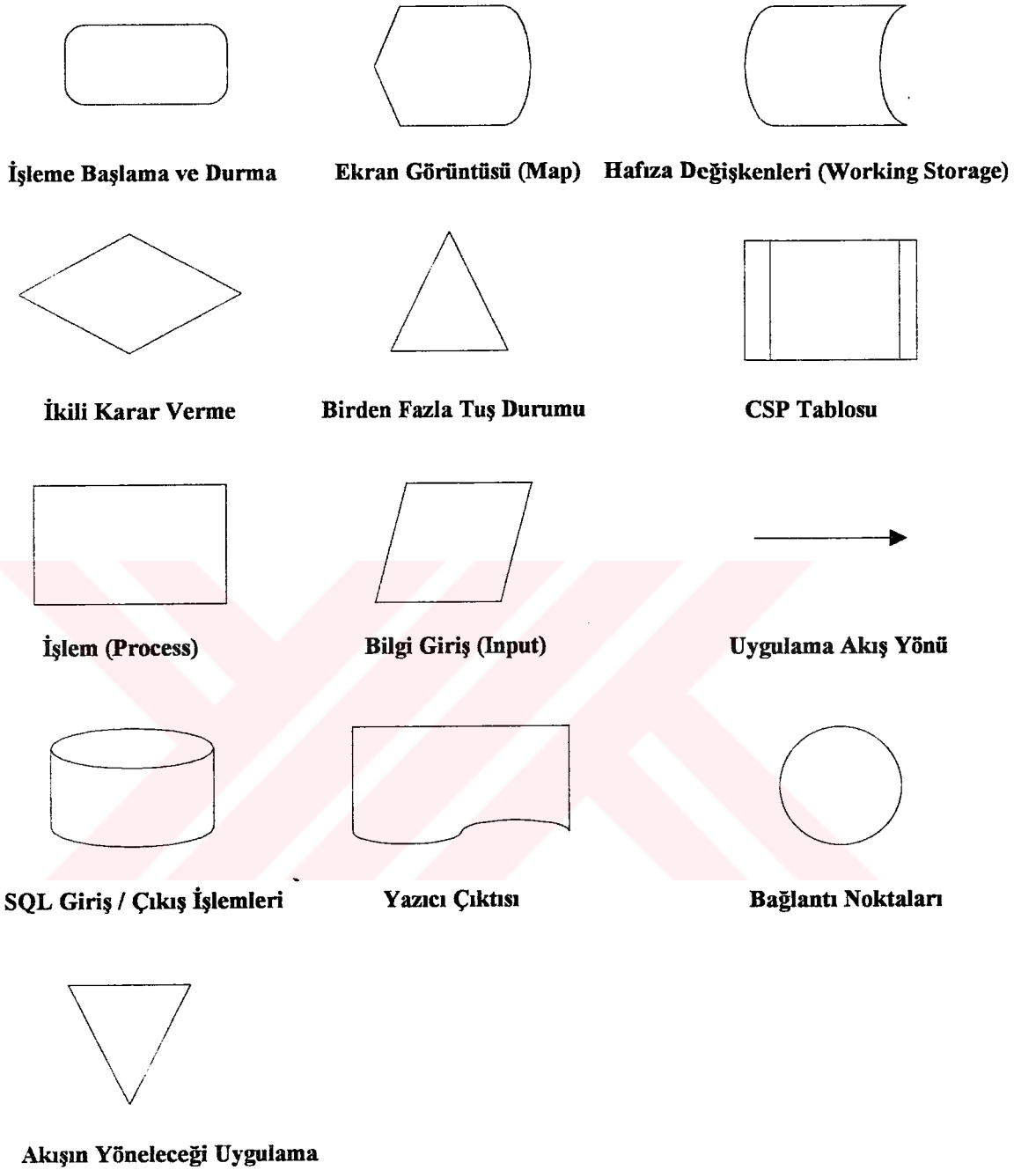
TA51M02	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Harita Bilgileri		
Aranan : Taraması yapılan bilgi ile alakalı olarak adet harita bulunmuştur.	Sayfa :		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Harita adı</td> </tr> <tr> <td style="height: 150px;"></td> </tr> </table>		Harita adı	
Harita adı			
PF2 : Çıkış	PF5 : Yazıcı		
PF7 : Geri	PF8 : İleri		

Ek 4.107. Harita Tarama Gelen Bilgi Paneli

TA51H01	ATATÜRK ÜNİVERSİTESİ Merkez Kütüphanesi Harita Bilgileri
<u>Açıklamalar :</u> 1- Taramak istediğiniz harita 1 / 25.000 ölçekli ve aranılan bölgede de “Erzurum” olsun. Bilgi alanına veri şu şekilde girilmelidir. A- Erzurum % : Erzurum kelimesi ile başlayan tüm kayıtları seçer. B- %Erzurum : Erzurum kelimesi ile biten kayıtların tümünü seçer. C- %Erzurum% : Erzurum kelimesi geçen tüm kayıtları seçer. 2- Elde edilen tarama sonuçları PF5 tuşuna basılarak yazıcıdan alınabilir.	
PF3 : Çıkış	

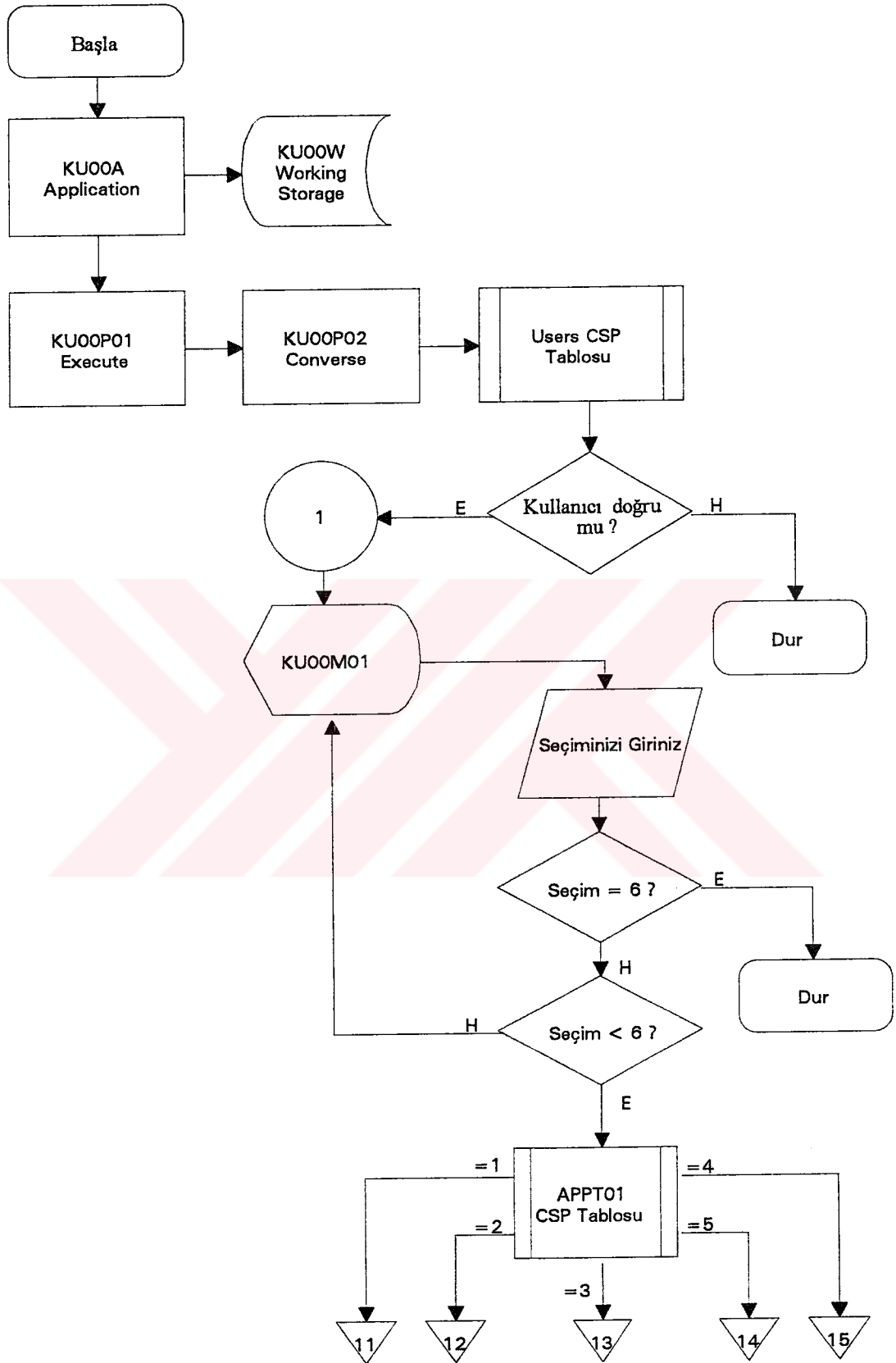
Ek 4.108. Harita Bilgileri Tarama Yardım Paneli

Ek 5. Uygulama Programı Akış Şemaları

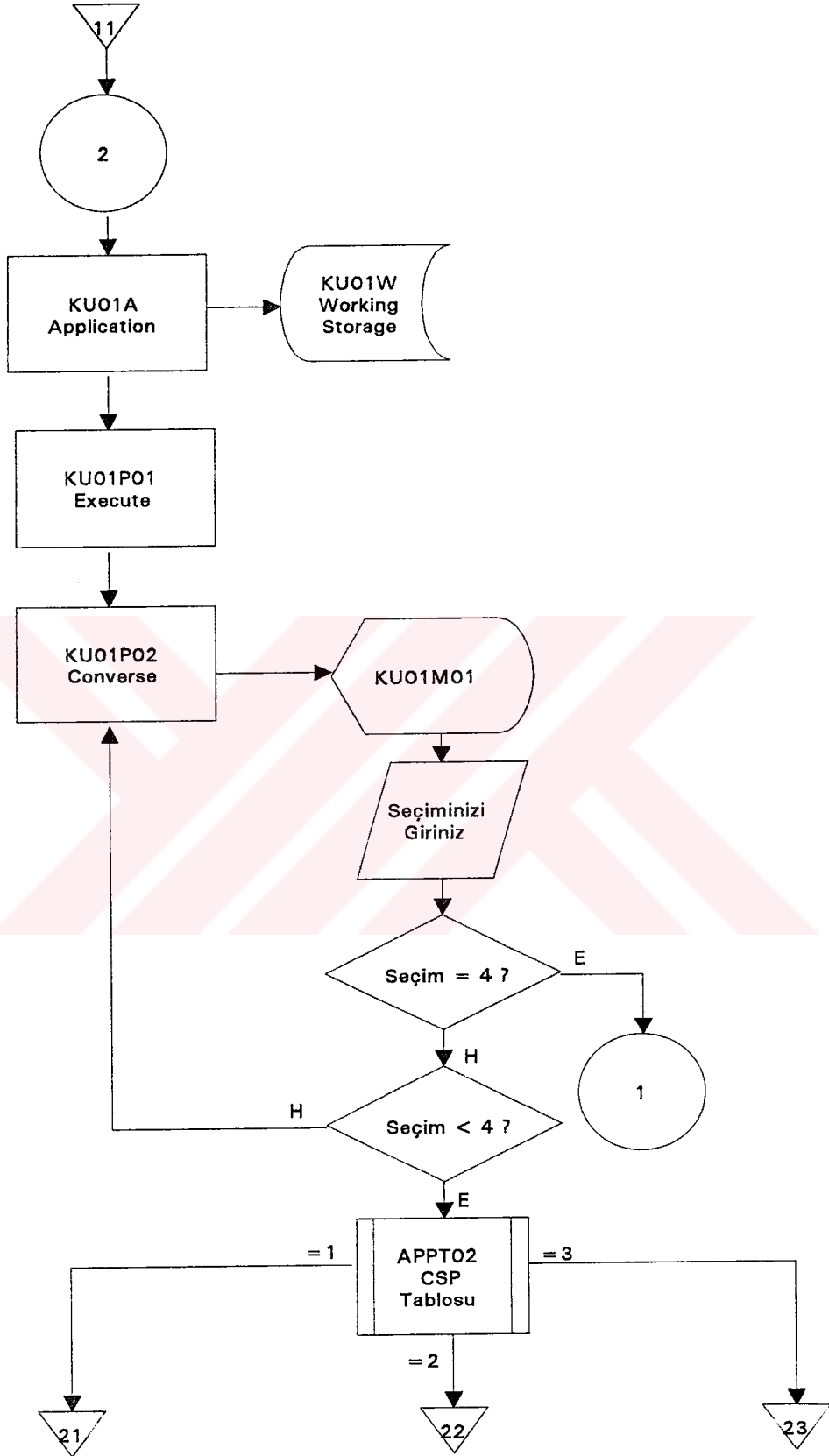


Not : Tüm ekranlarda Pf1 tuşu yardım (help) tuşu olarak düzenlenmiştir. Pf7 ekranlarda bir önceki sayfaya, Pf8 tuşu ise bir sonraki sayfaya erişimi sağlayan tuşlardır. Akış şemalarında Statement Group'lar gösterilmemiştir.

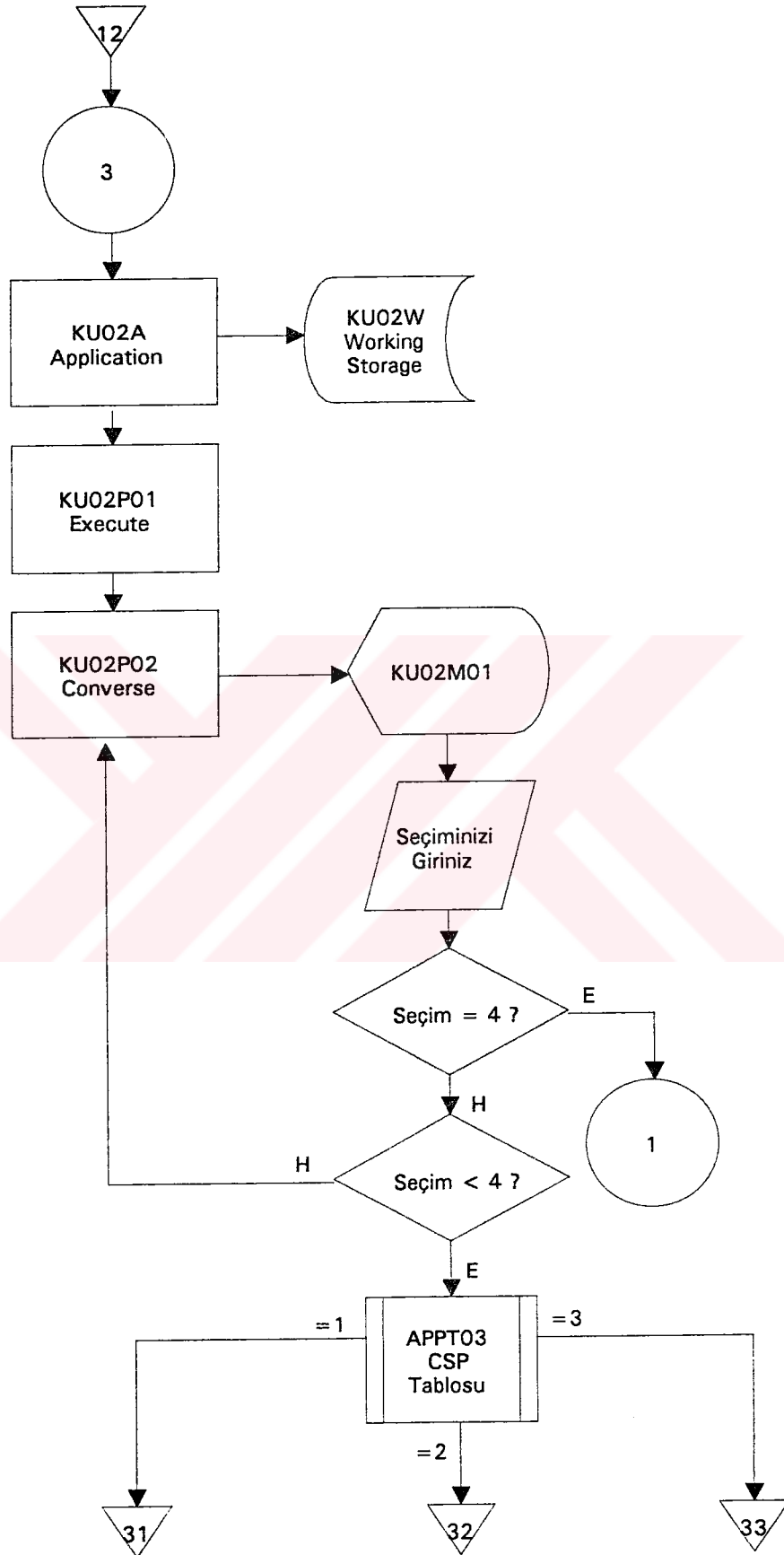
Ek 5.1. Akış Şemalarında Kullanılan Semboller



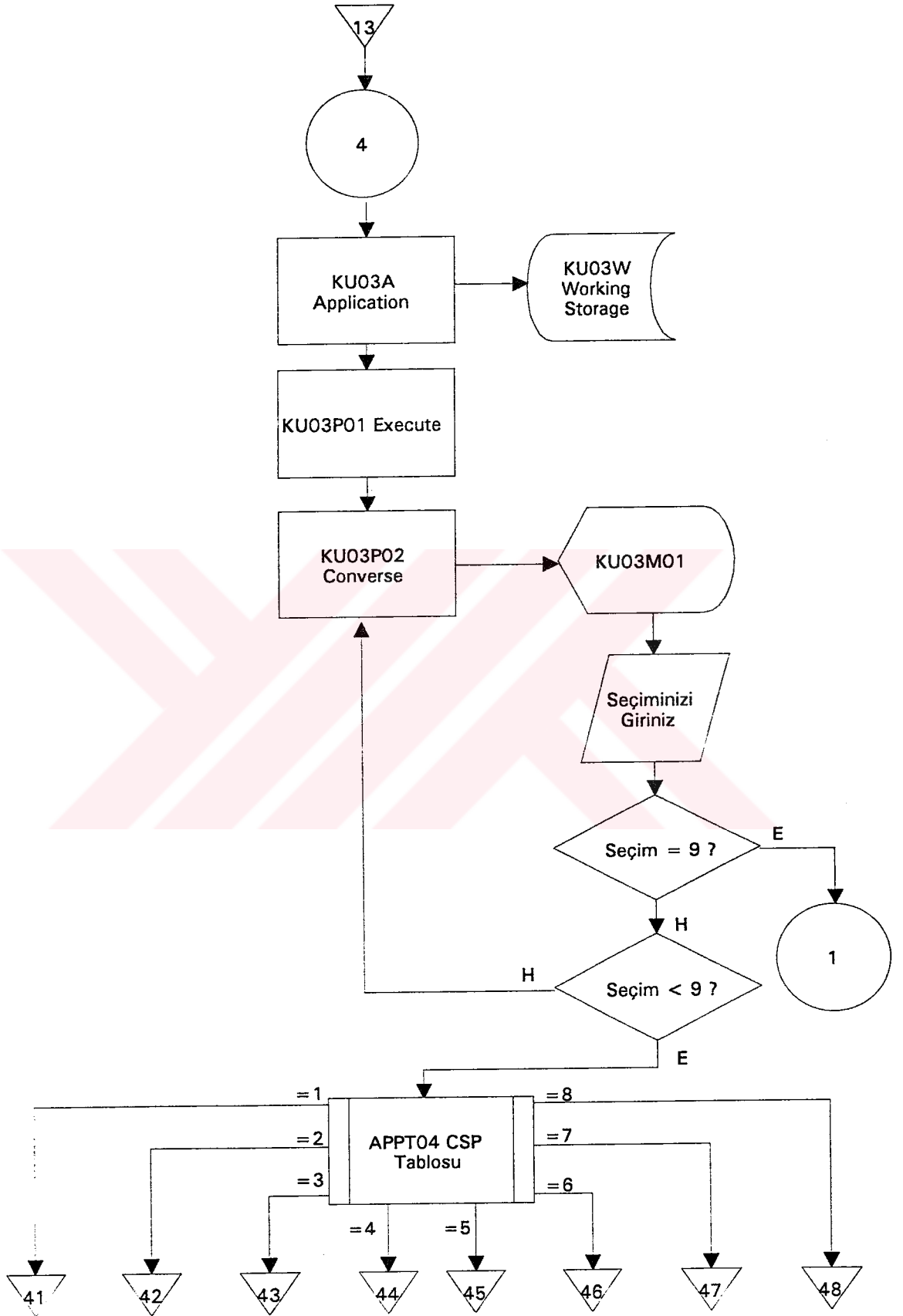
Ek 5.2. KU00A Uygulama Akış Şeması



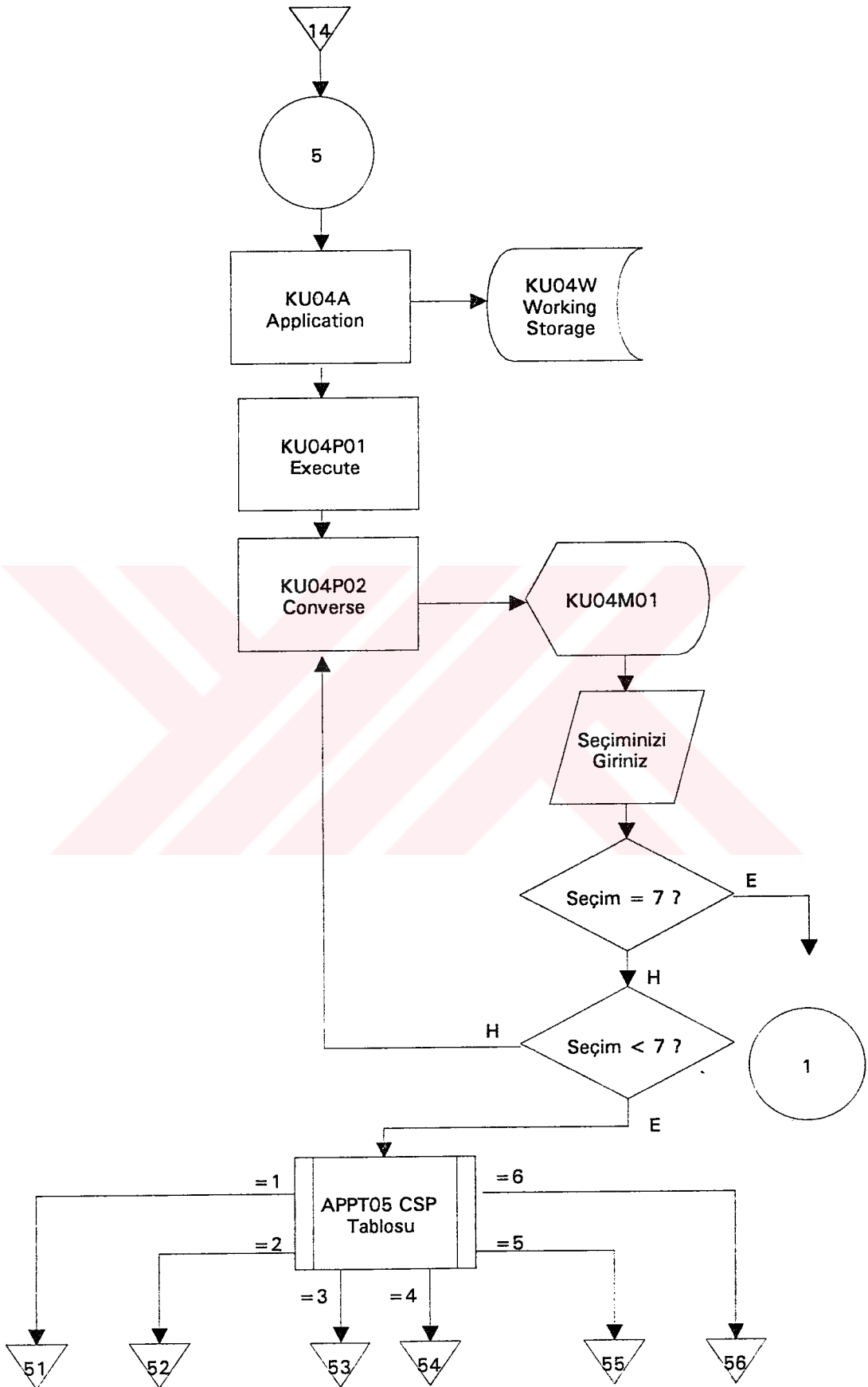
Ek 5.3. KU01A Uygulama Akış Şeması



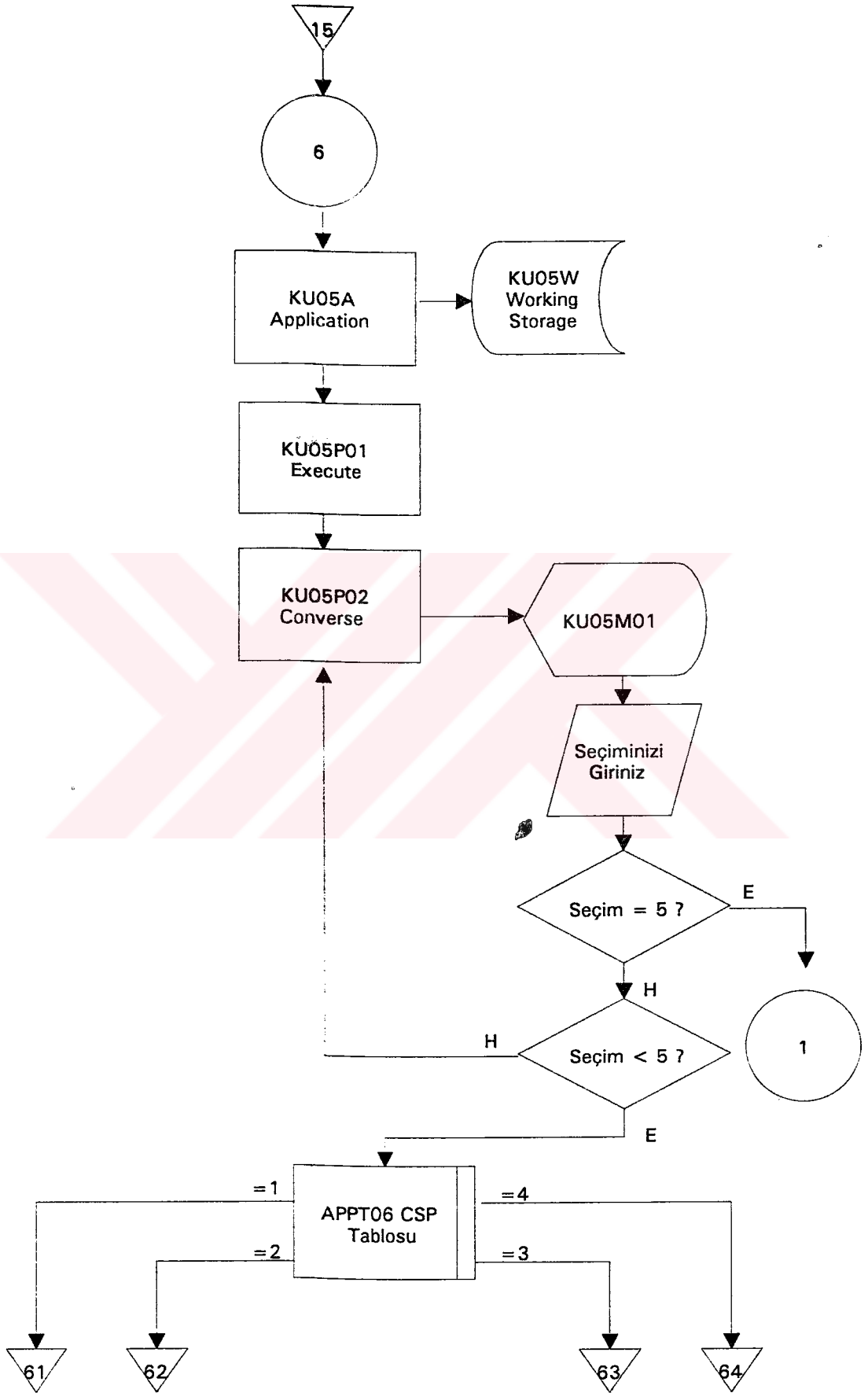
Ek 5.4. KU02A Uygulama Akış şeması



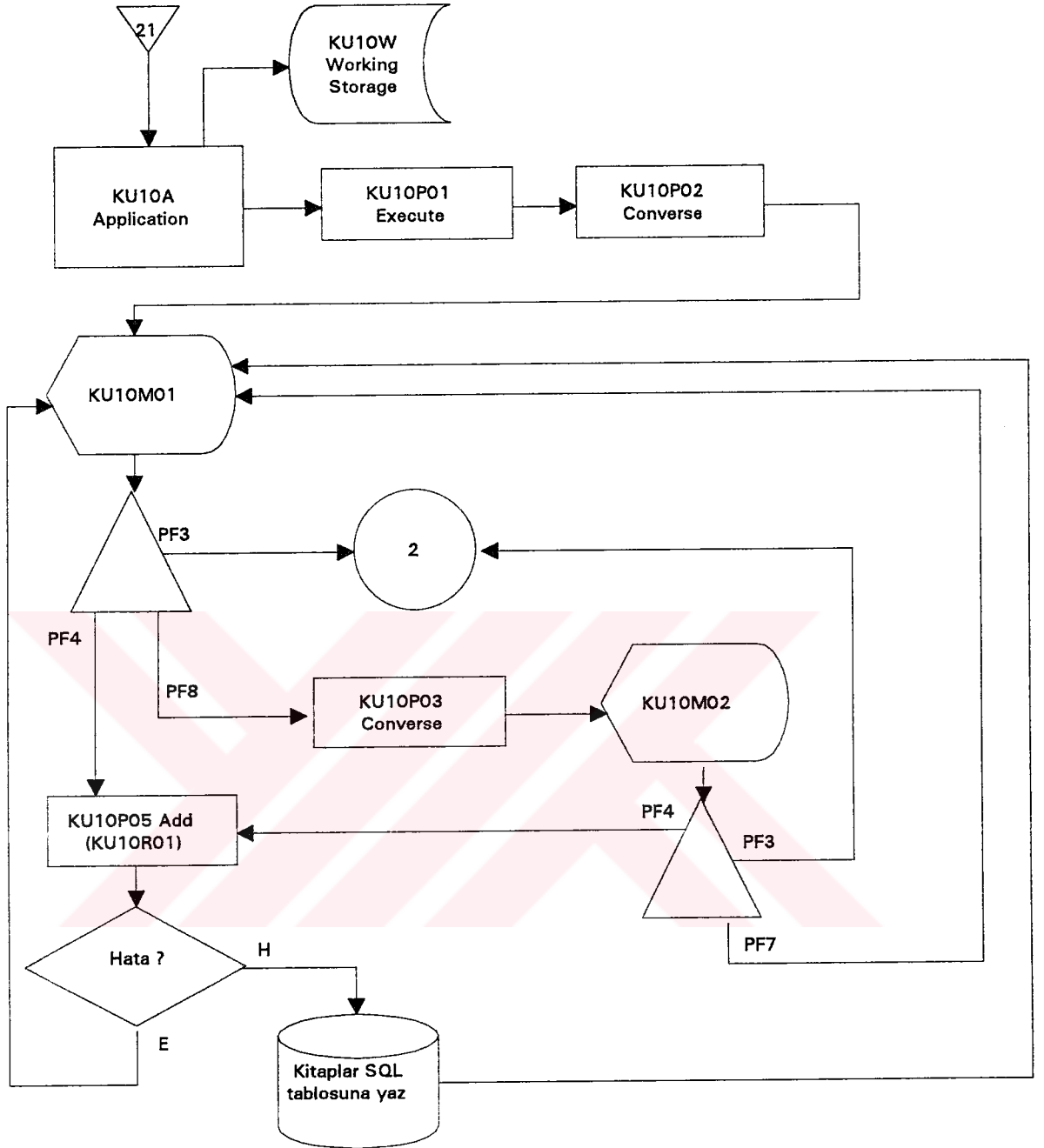
Ek 5.5. KU03A Uygulama Akış Şeması



Ek 5.6. KU04A Uygulama Akış Şeması

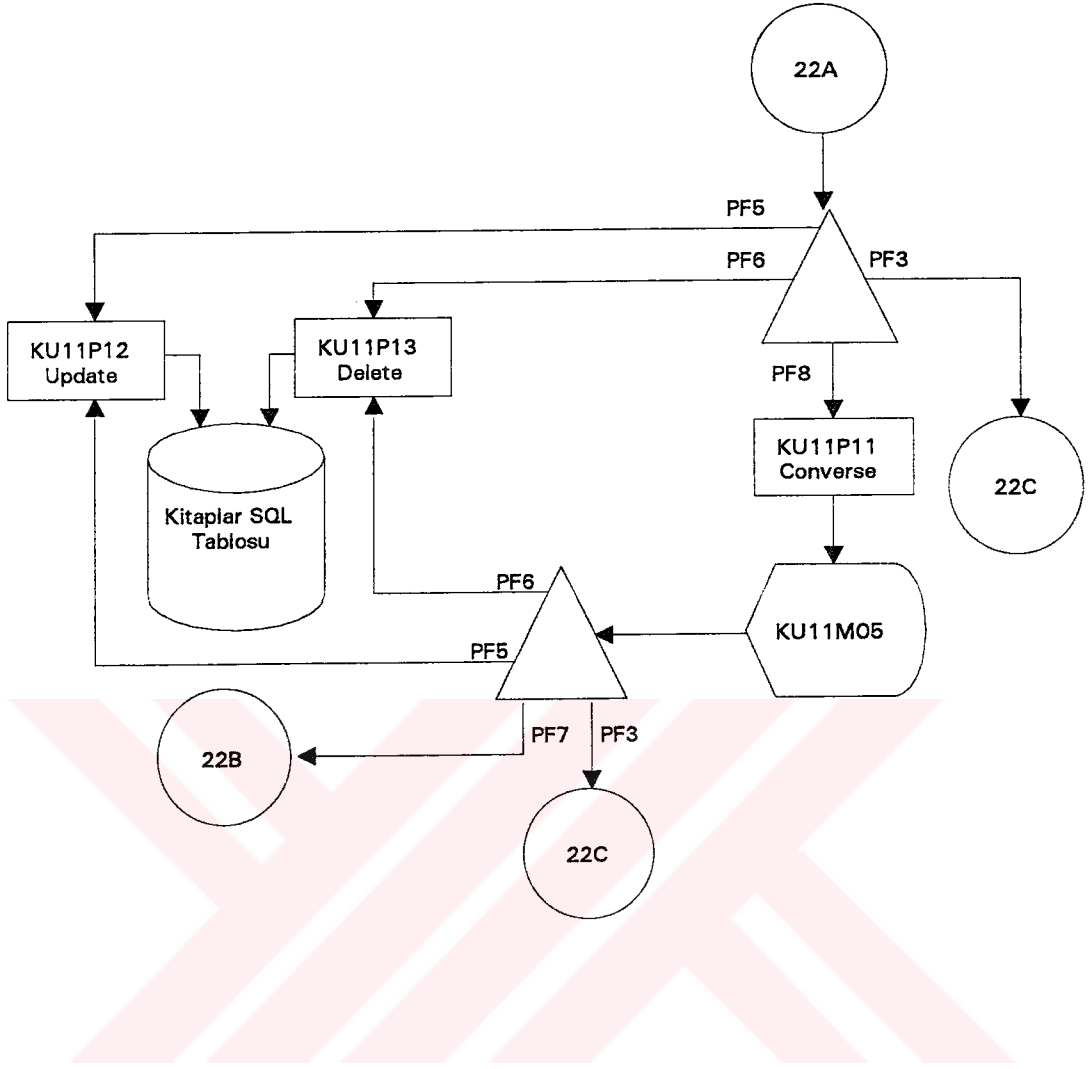


Ek 5.7. KU05A Uygulama Akış Şeması



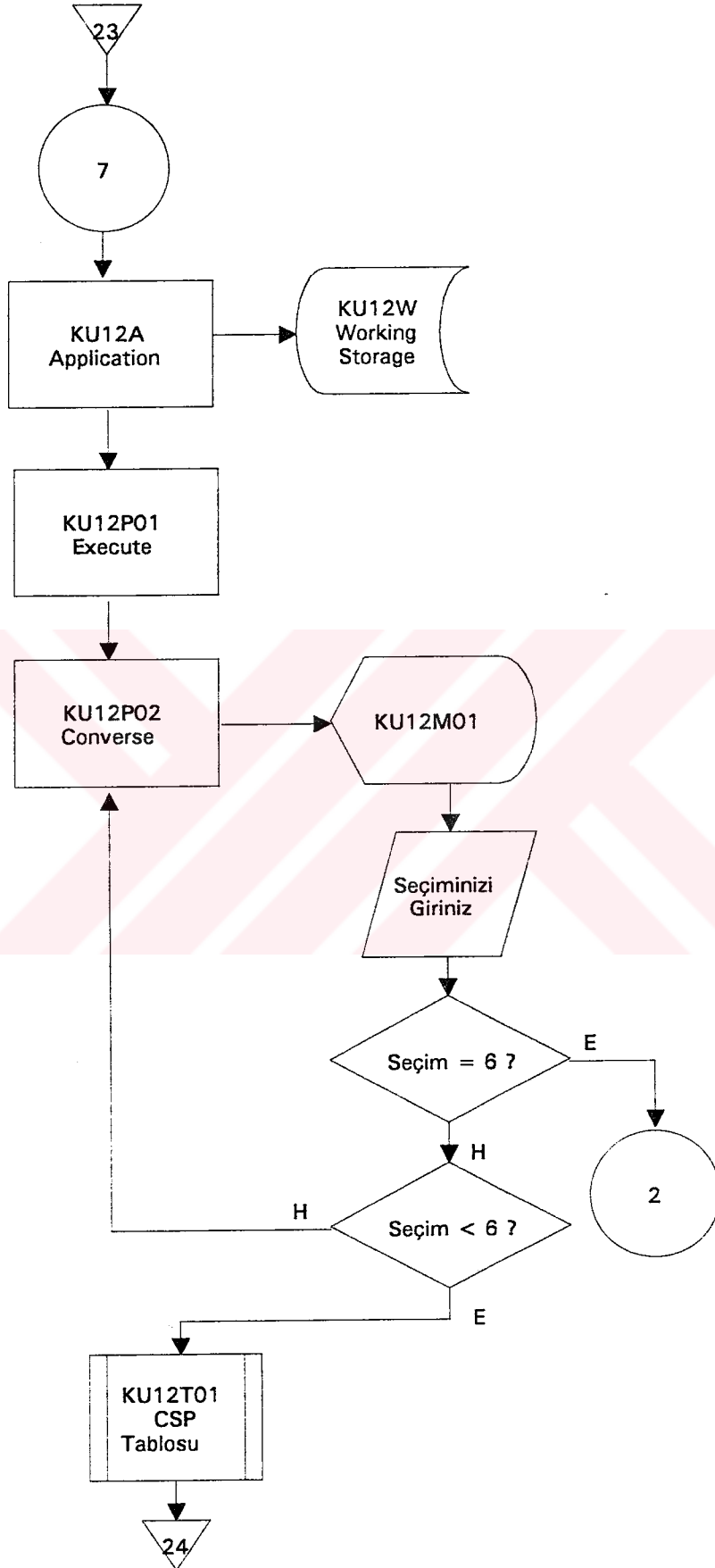
Ek 5.8. KU10A Uygulama Akış Şeması

Ek 5.9. KU11A Uygulama Akış Şeması

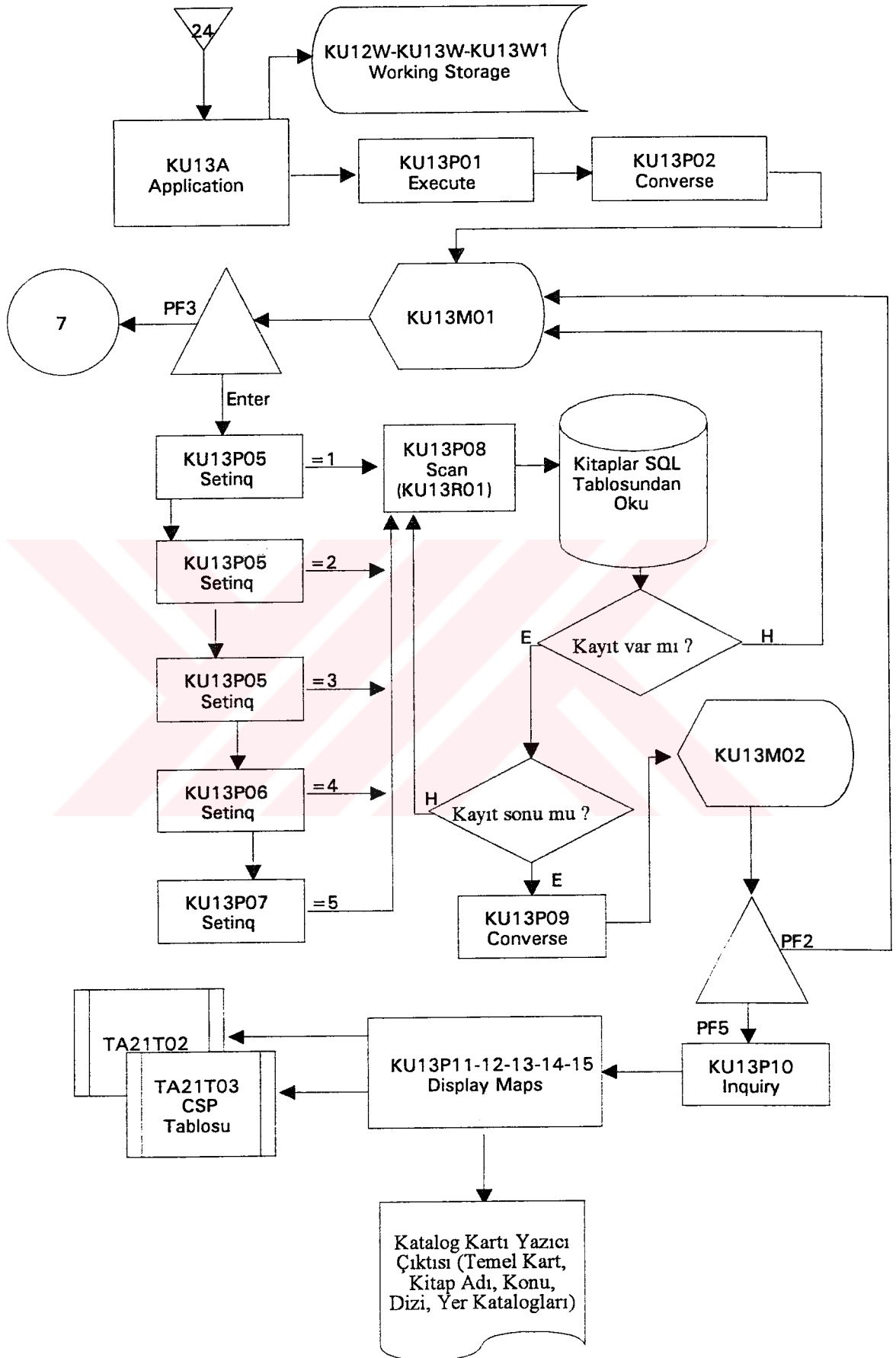


Ek 5.9. KU11A Uygulama Akış Şeması (Devam)

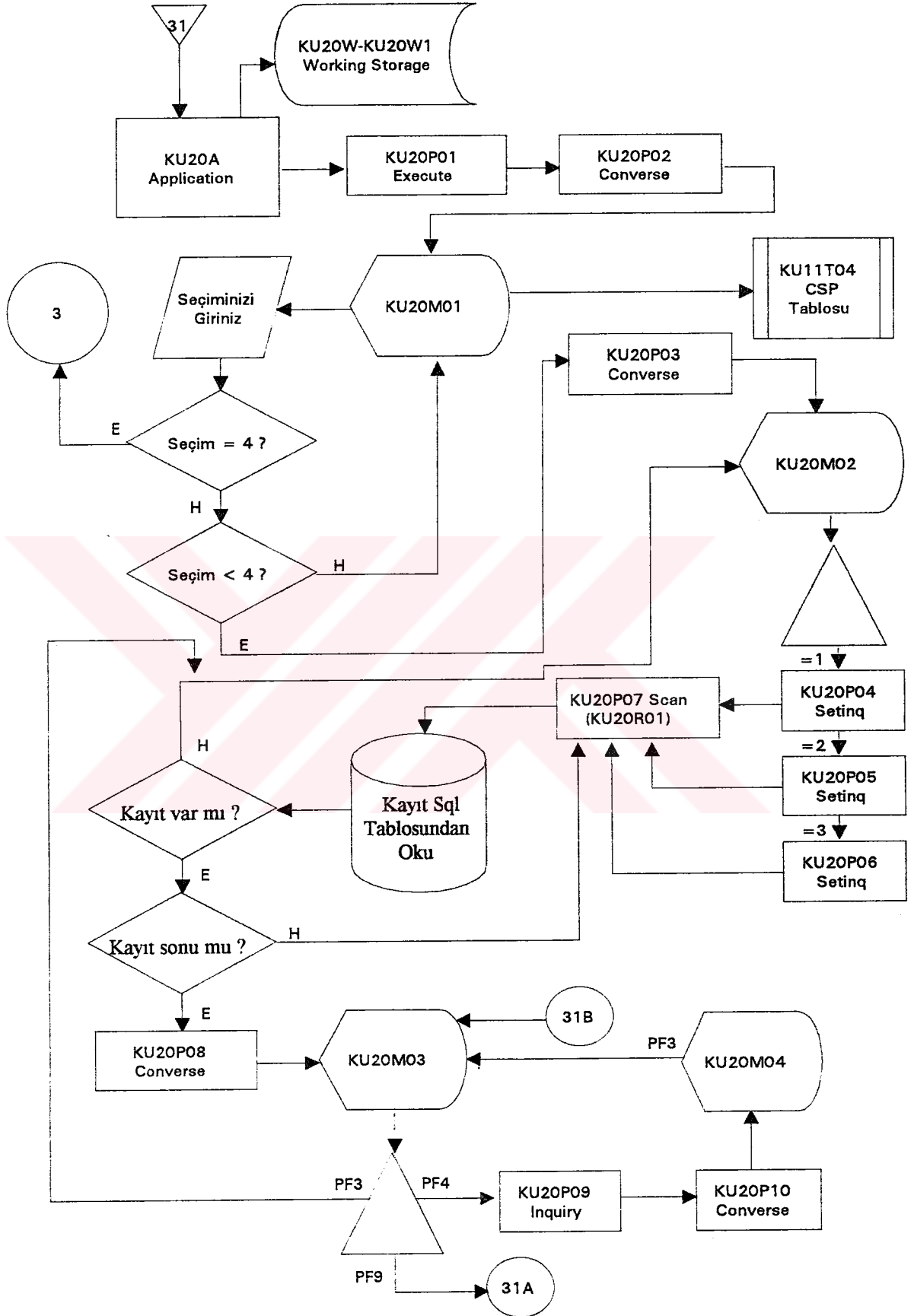
AKINCO
ARBAŞI



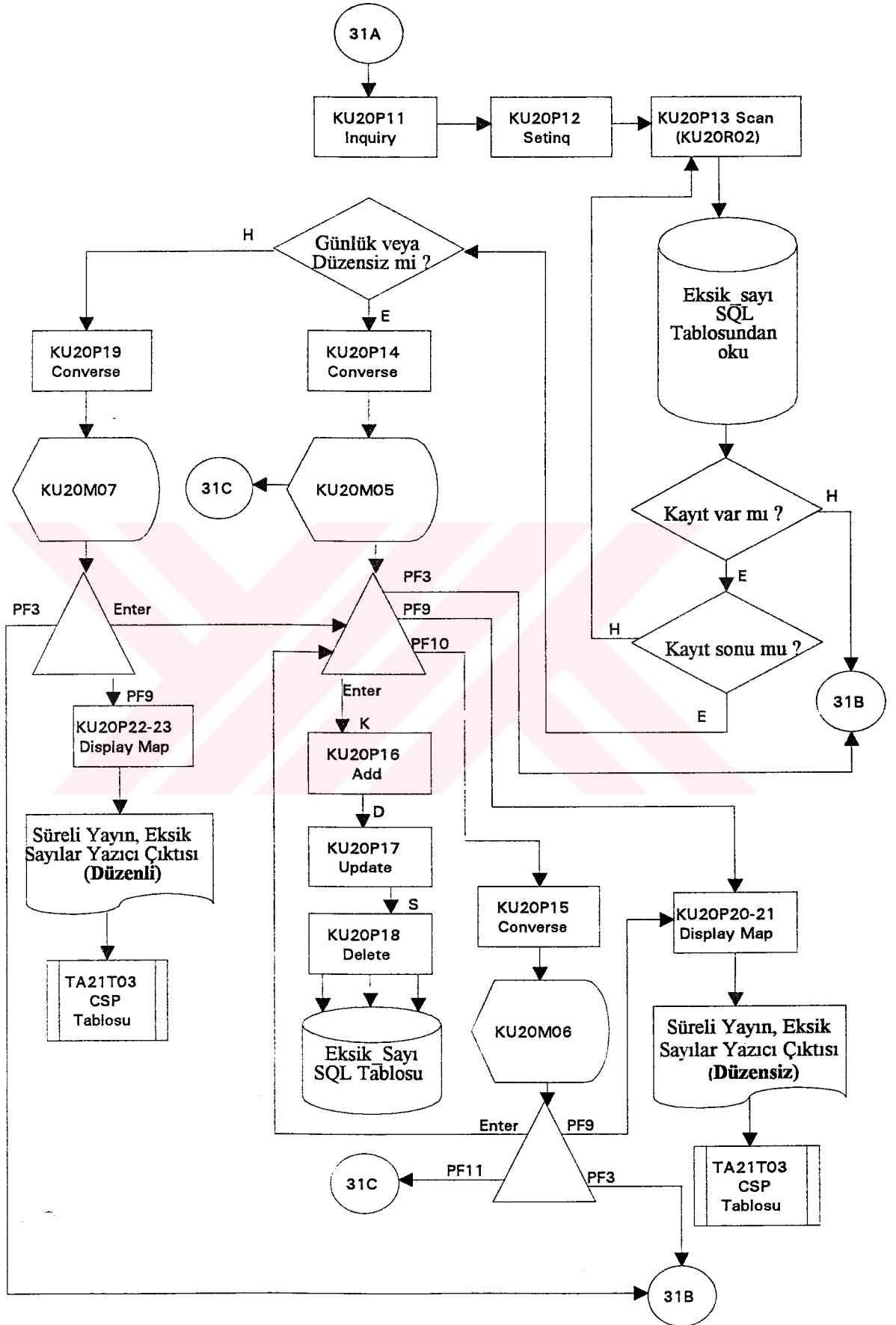
Ek 5.10. KU12A Uygulama Akış Şeması



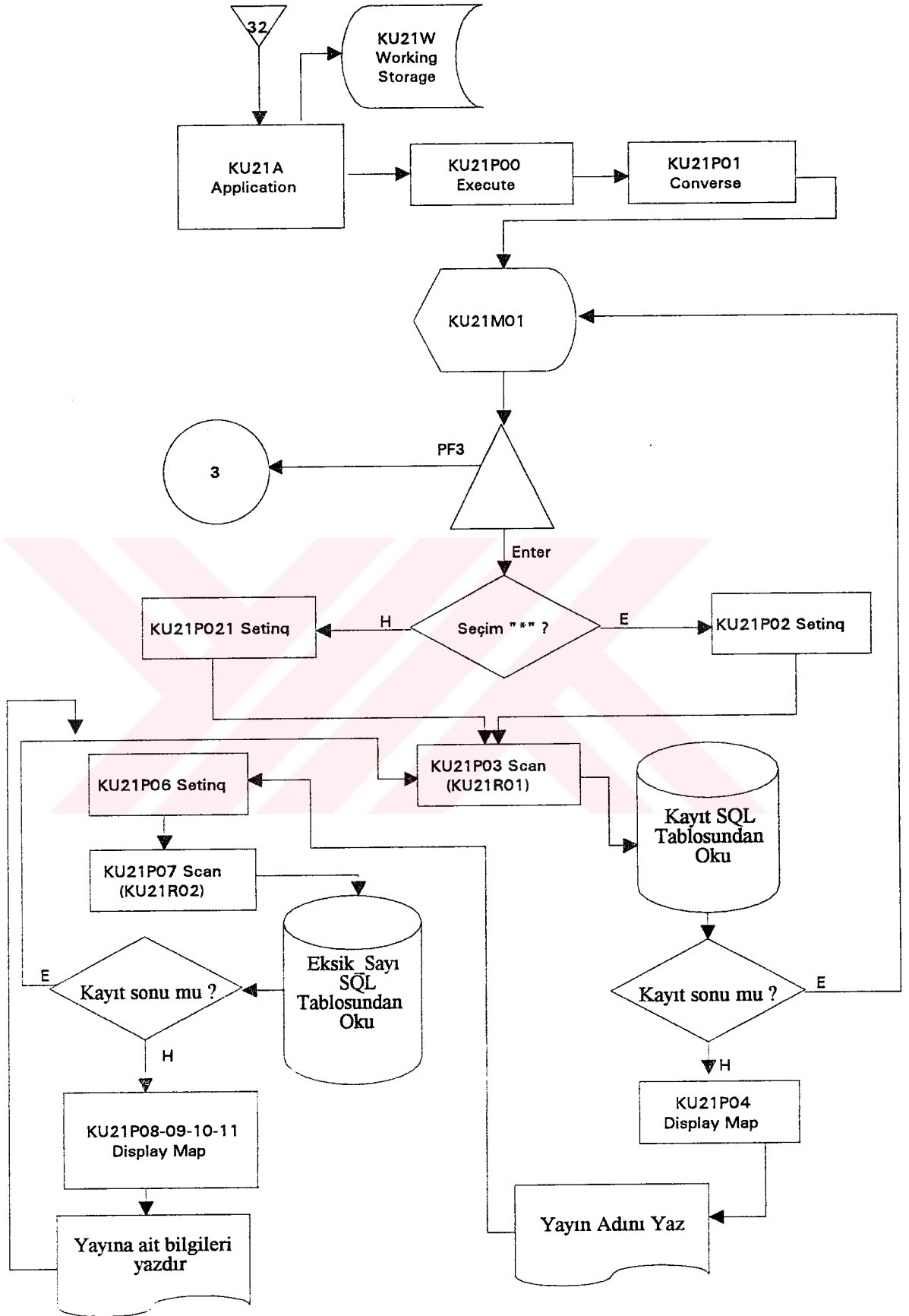
Ek 5.11. KU13A Uygulama Akış Şeması



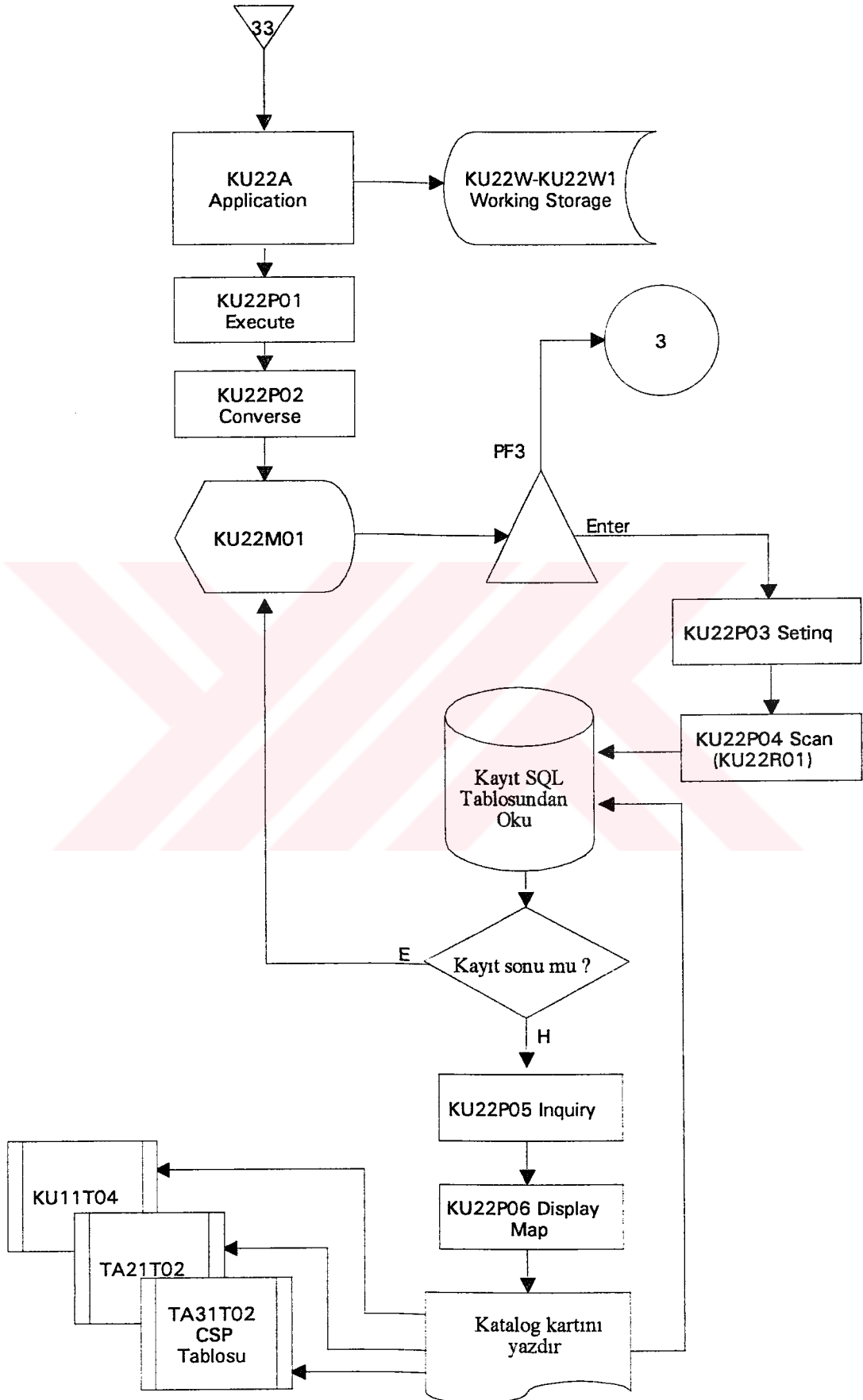
Ek 5.12. KU20A Uygulama Akış Şeması



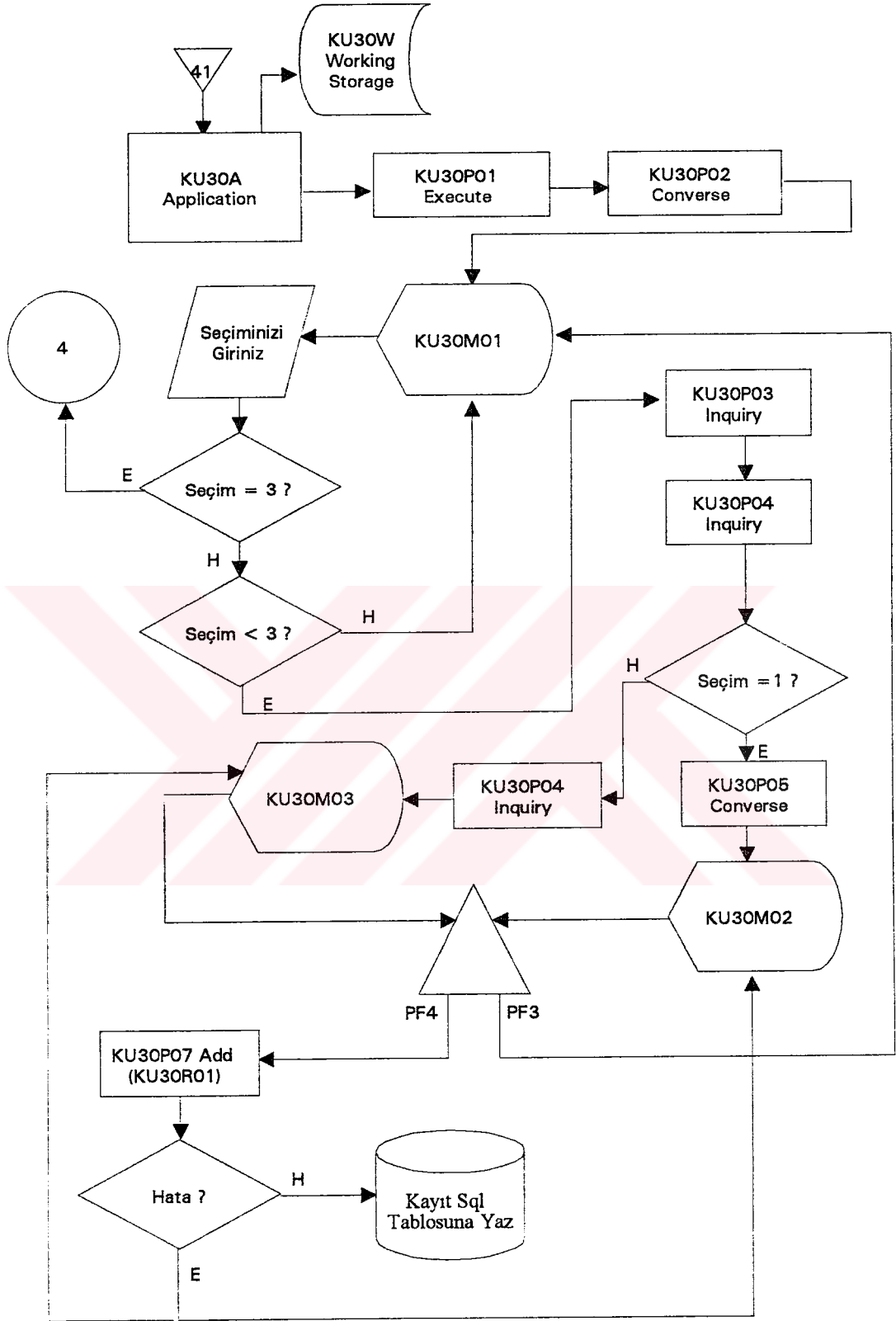
Ek 5.12. KU20A Uygulama Akış Şeması (Devam)



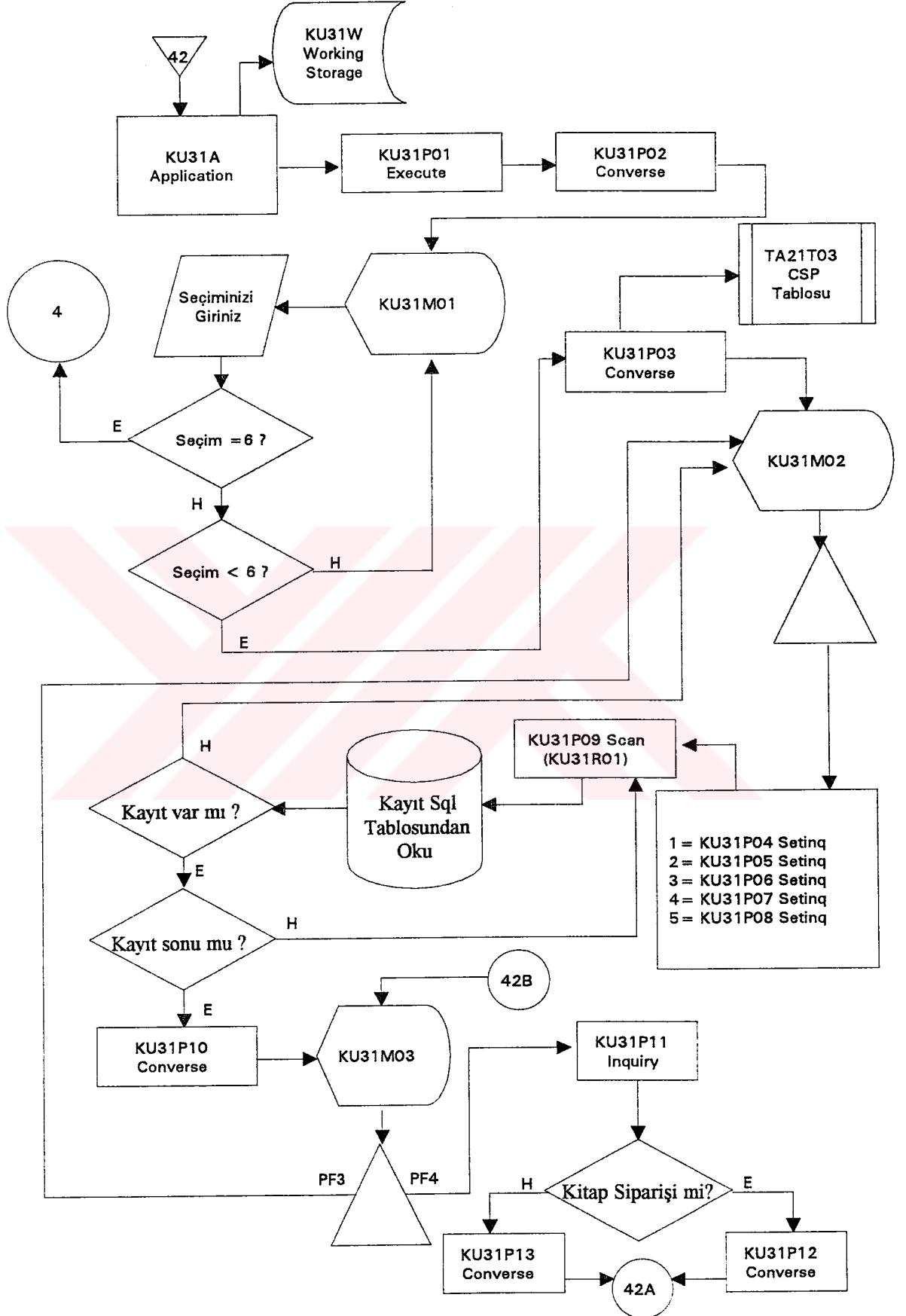
Ek 5.13. KU21A Uygulama Akış Şeması



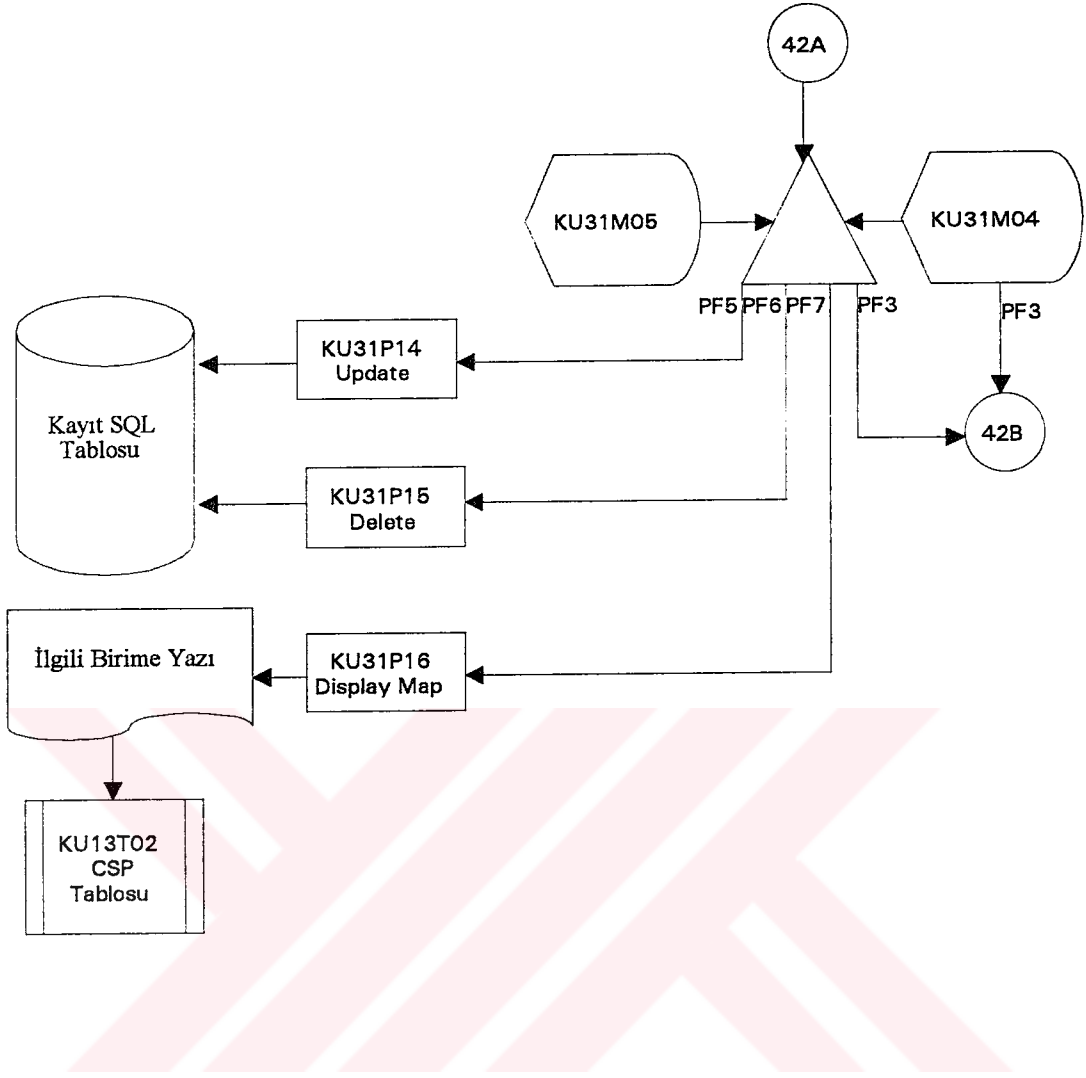
Ek 5.14. KU22.A Uygulama Akış Şeması



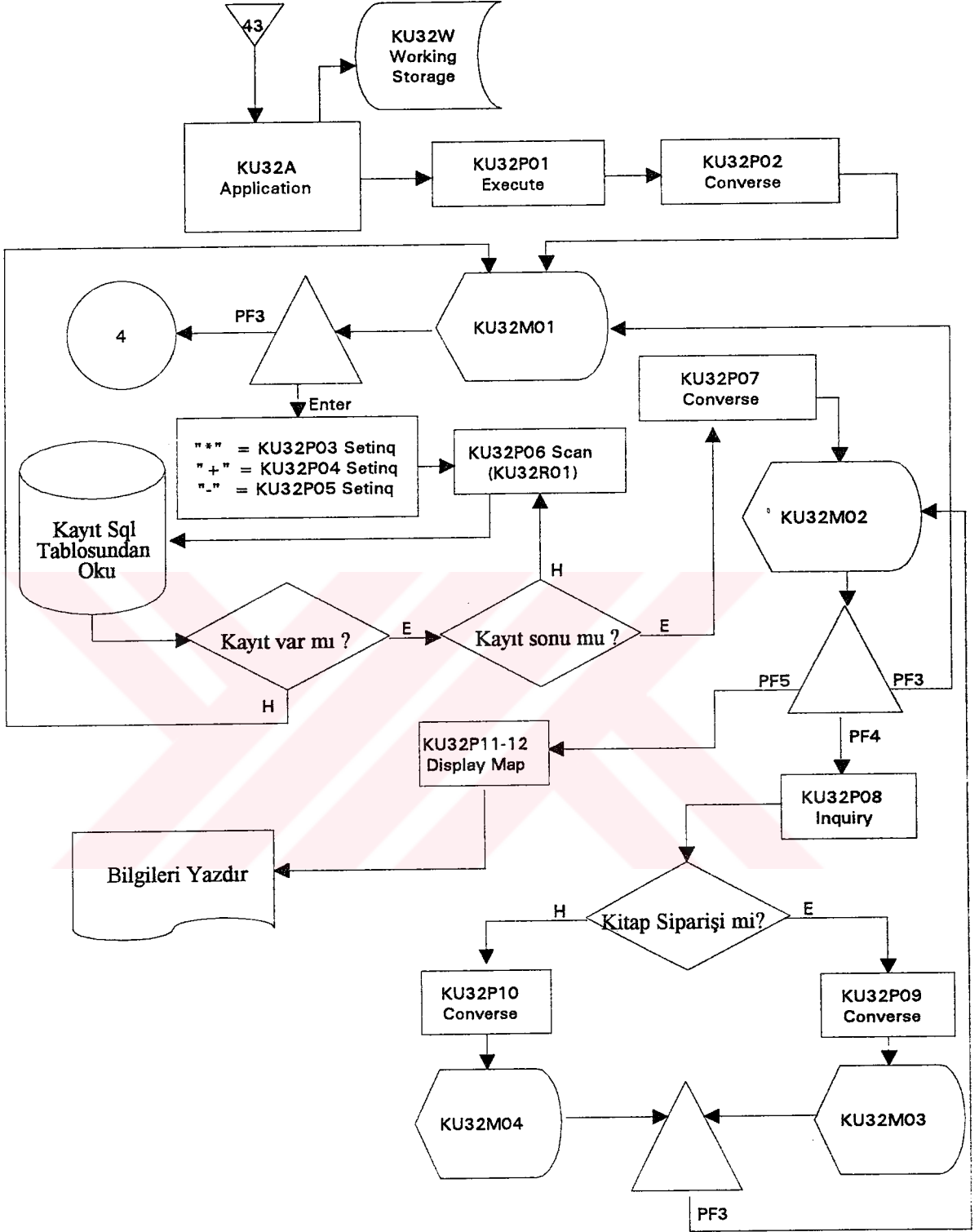
Ek 5.15. KU30A Uygulama Akış Şeması



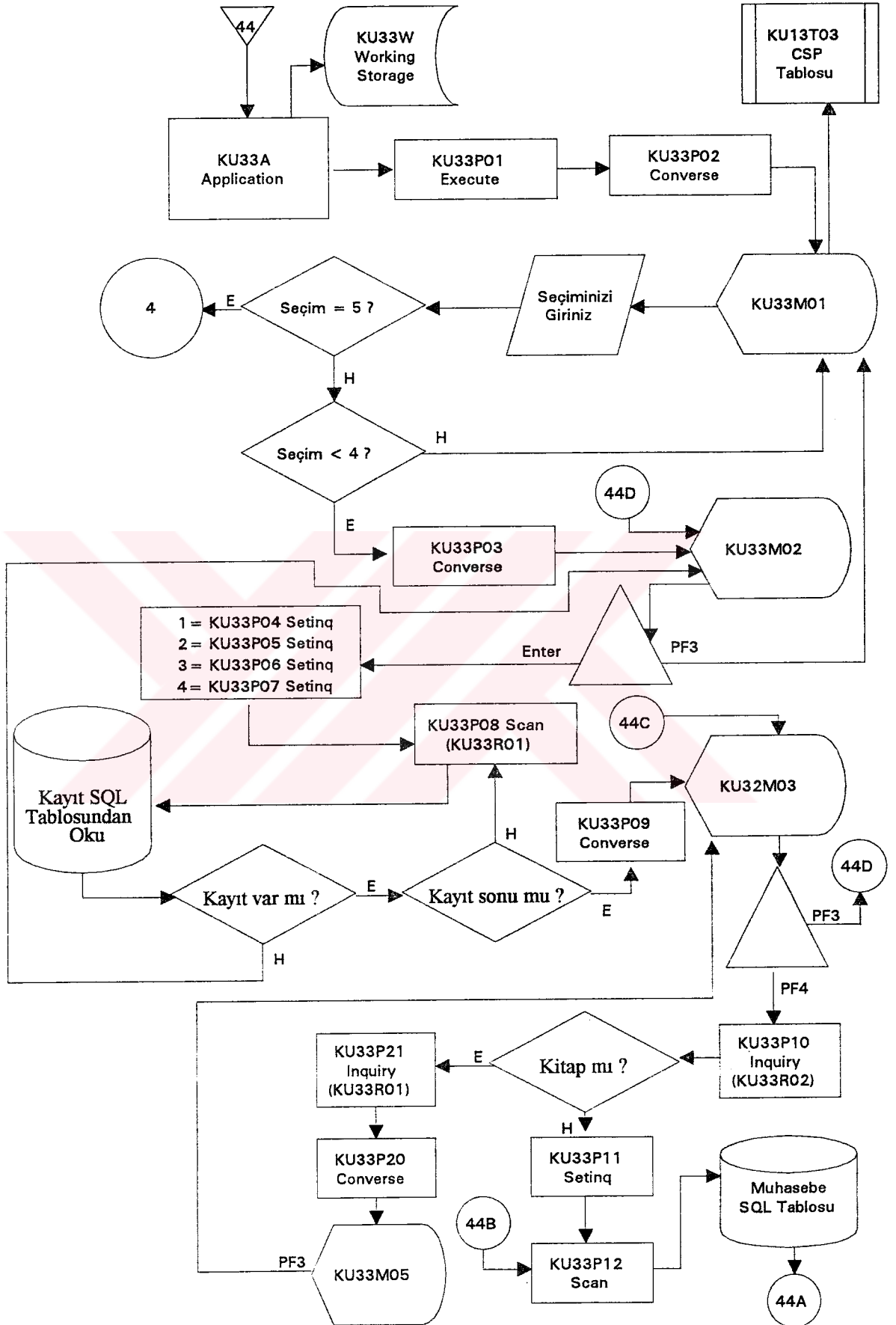
Ek 5.16. KU31A Uygulama Akış Şeması



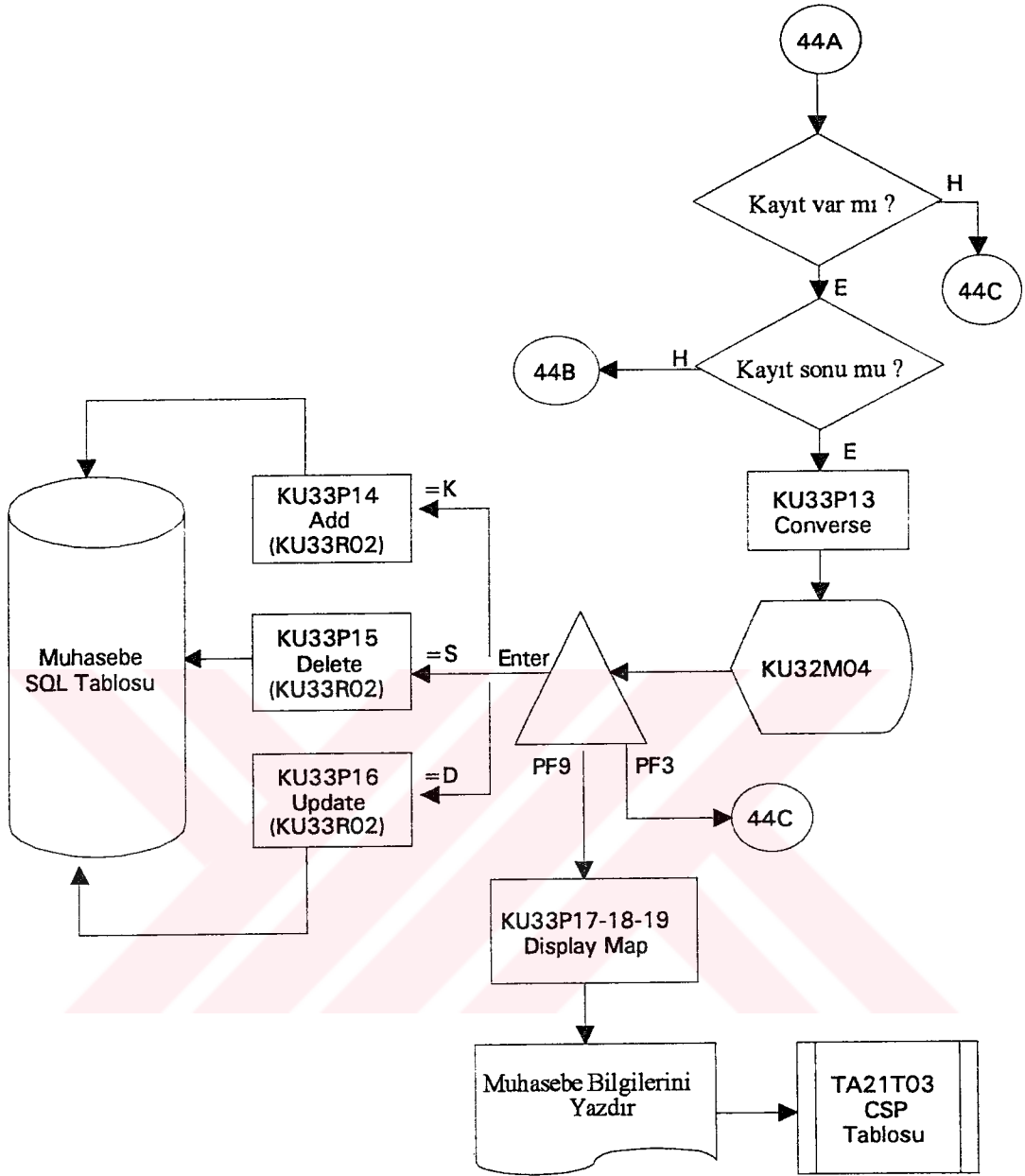
Ek 5.16. KU31A Uygulama Akış Şeması (Devam)



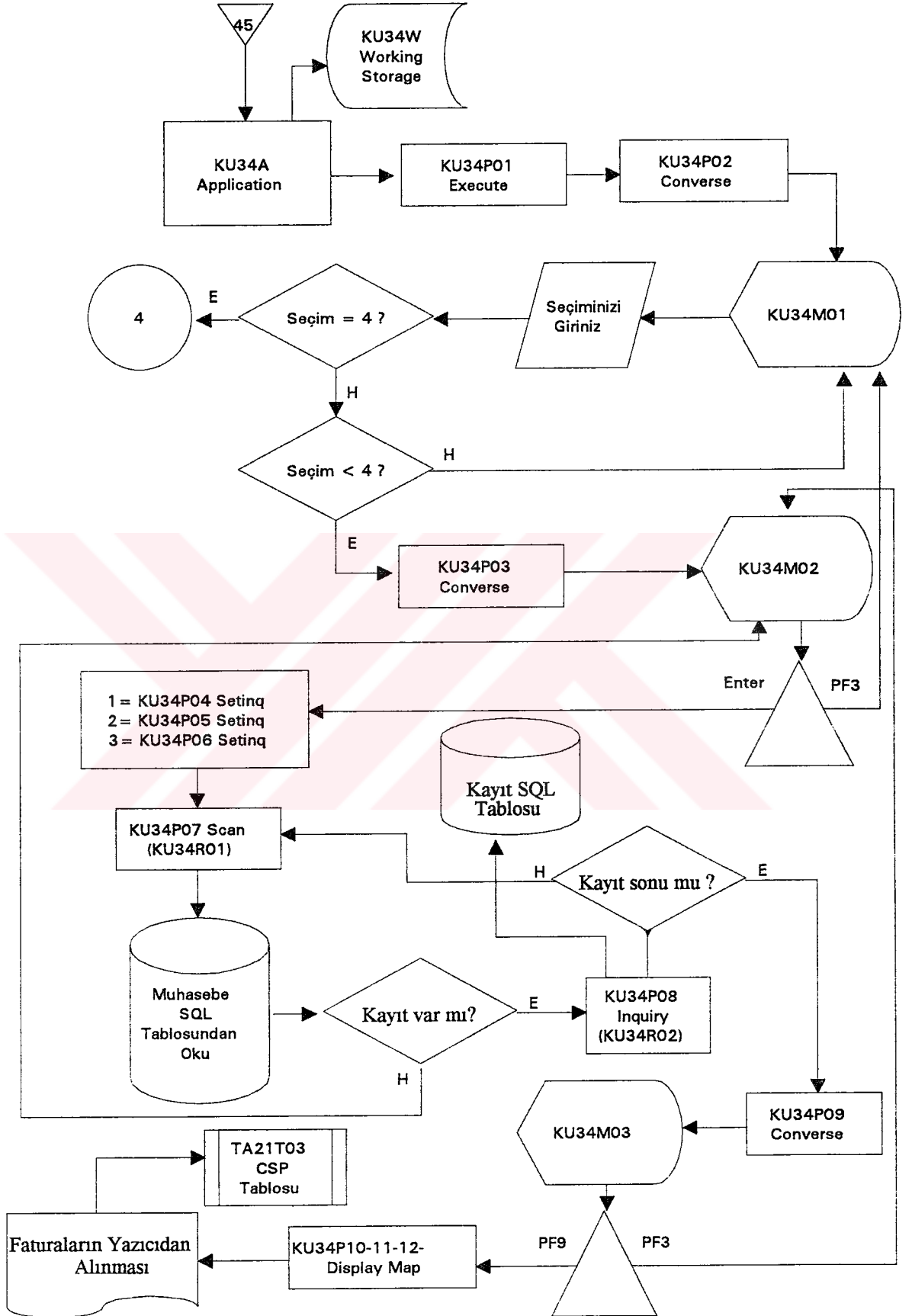
Ek 5.17. KU32A Uygulama Akış Şeması



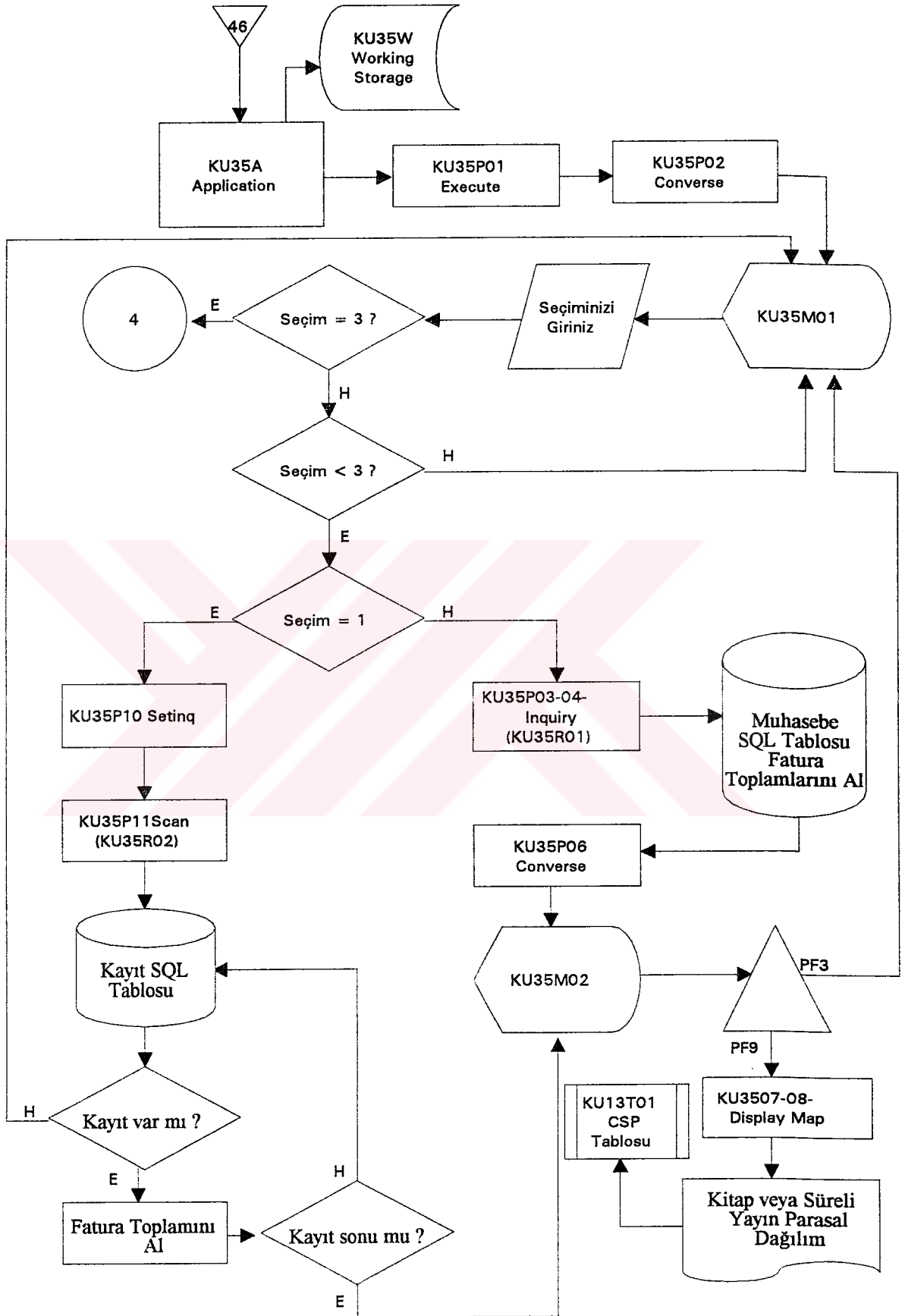
Ek 5.18. KU33A Uygulama Akış Şeması



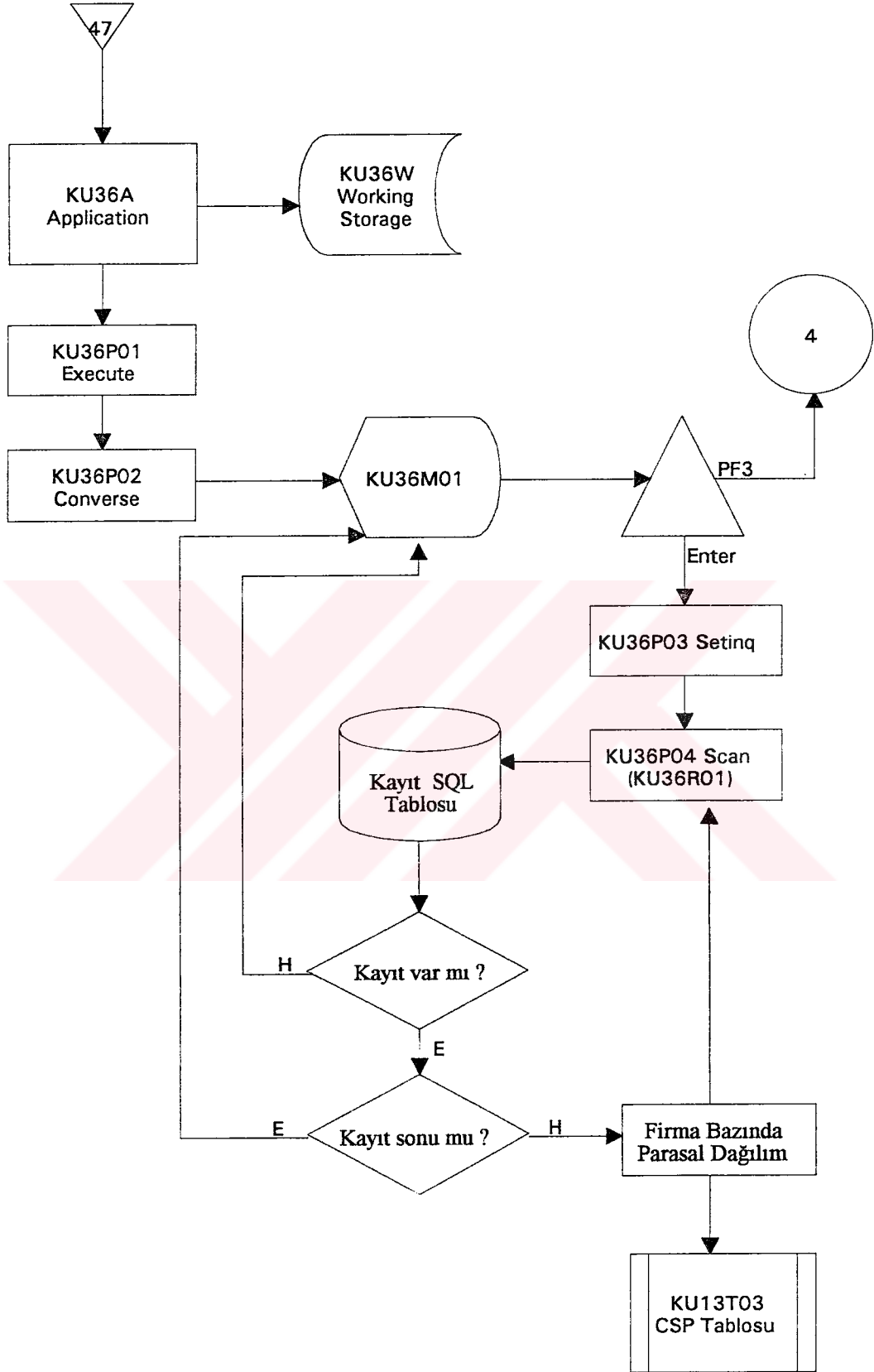
Ek 5.18. KU33A Uygulama Akış Şeması (Devam)



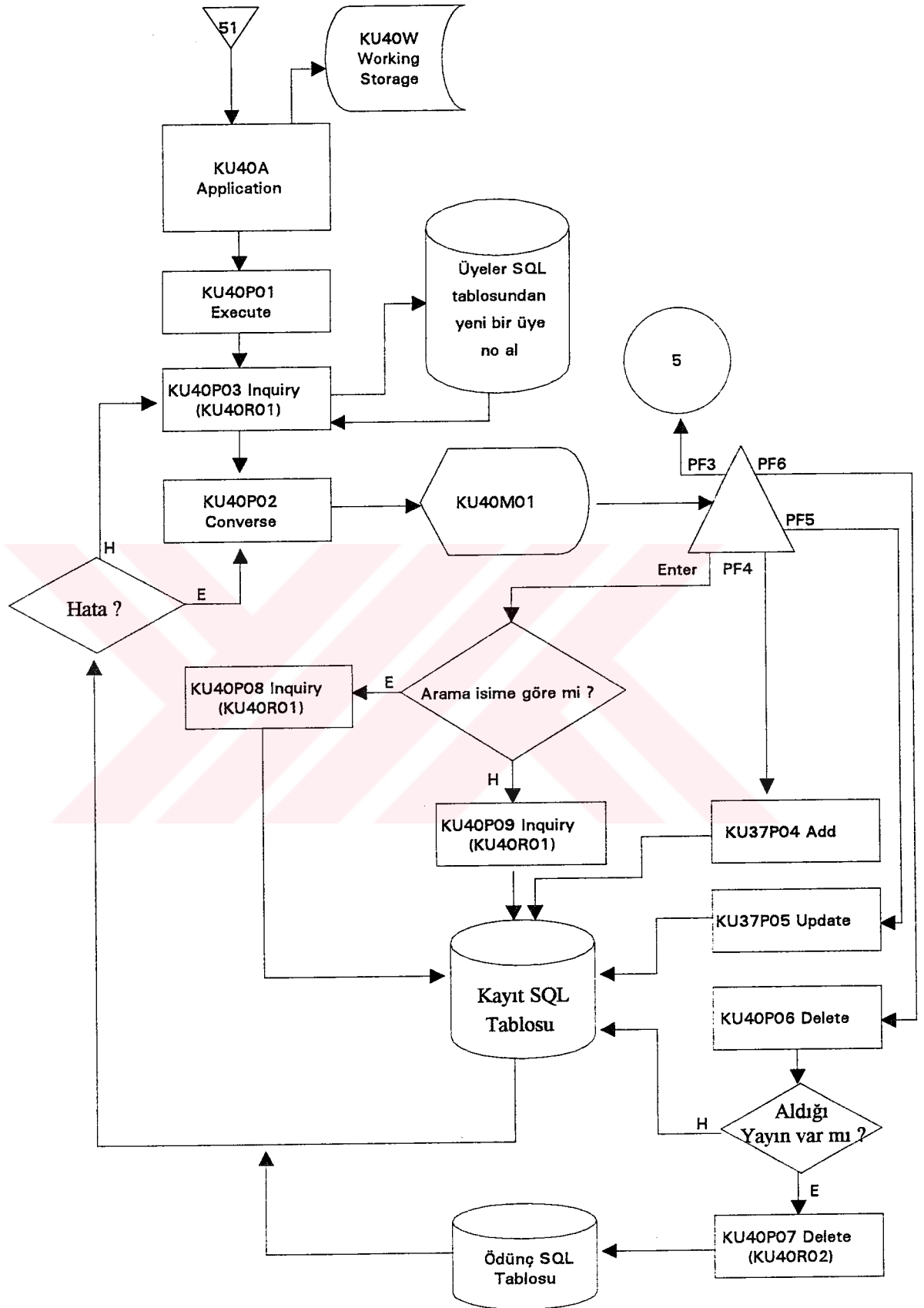
Ek 5.19. KU34A Uygulama Akış Şeması



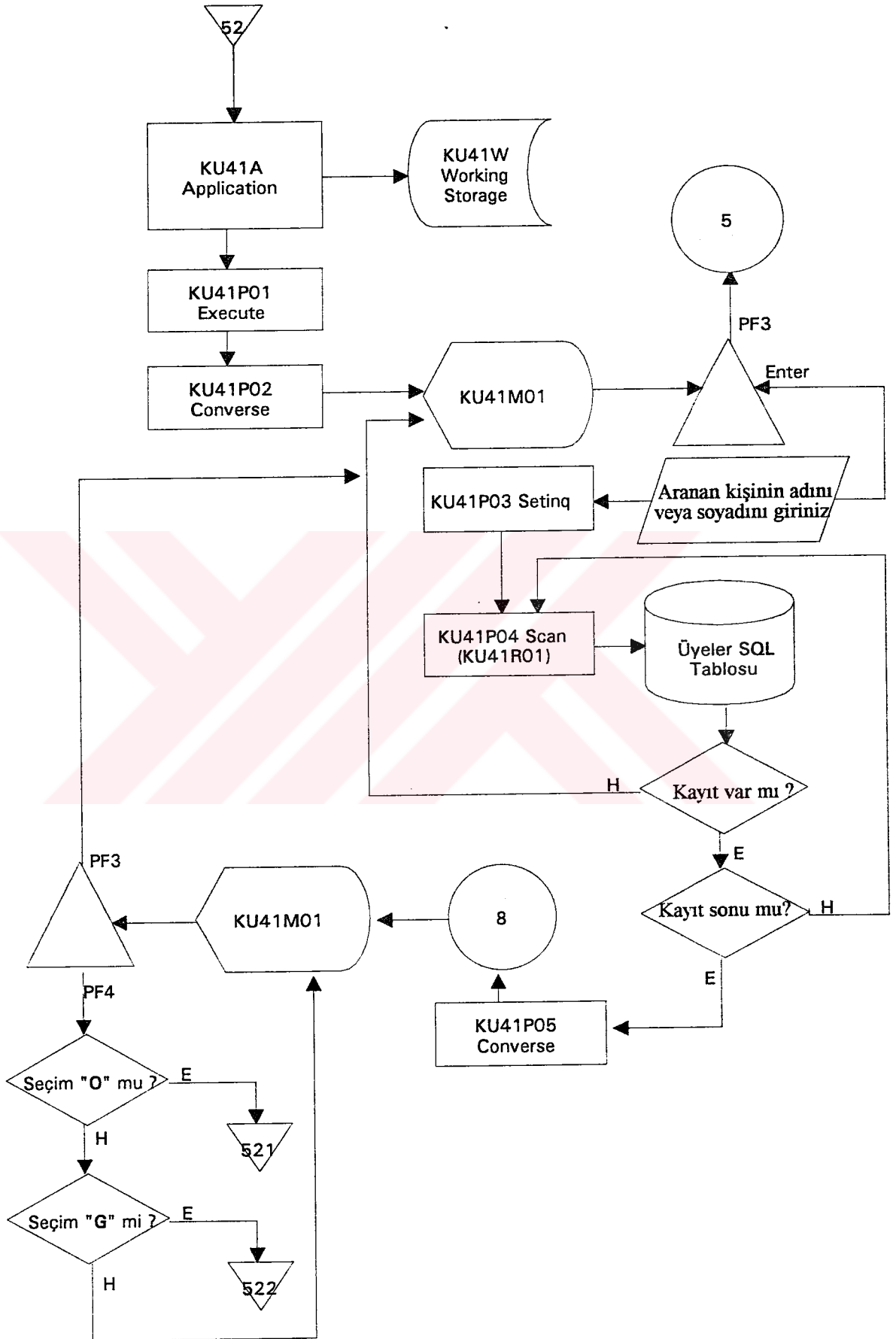
Ek 5.20. KU35A Uygulama Akış Şeması



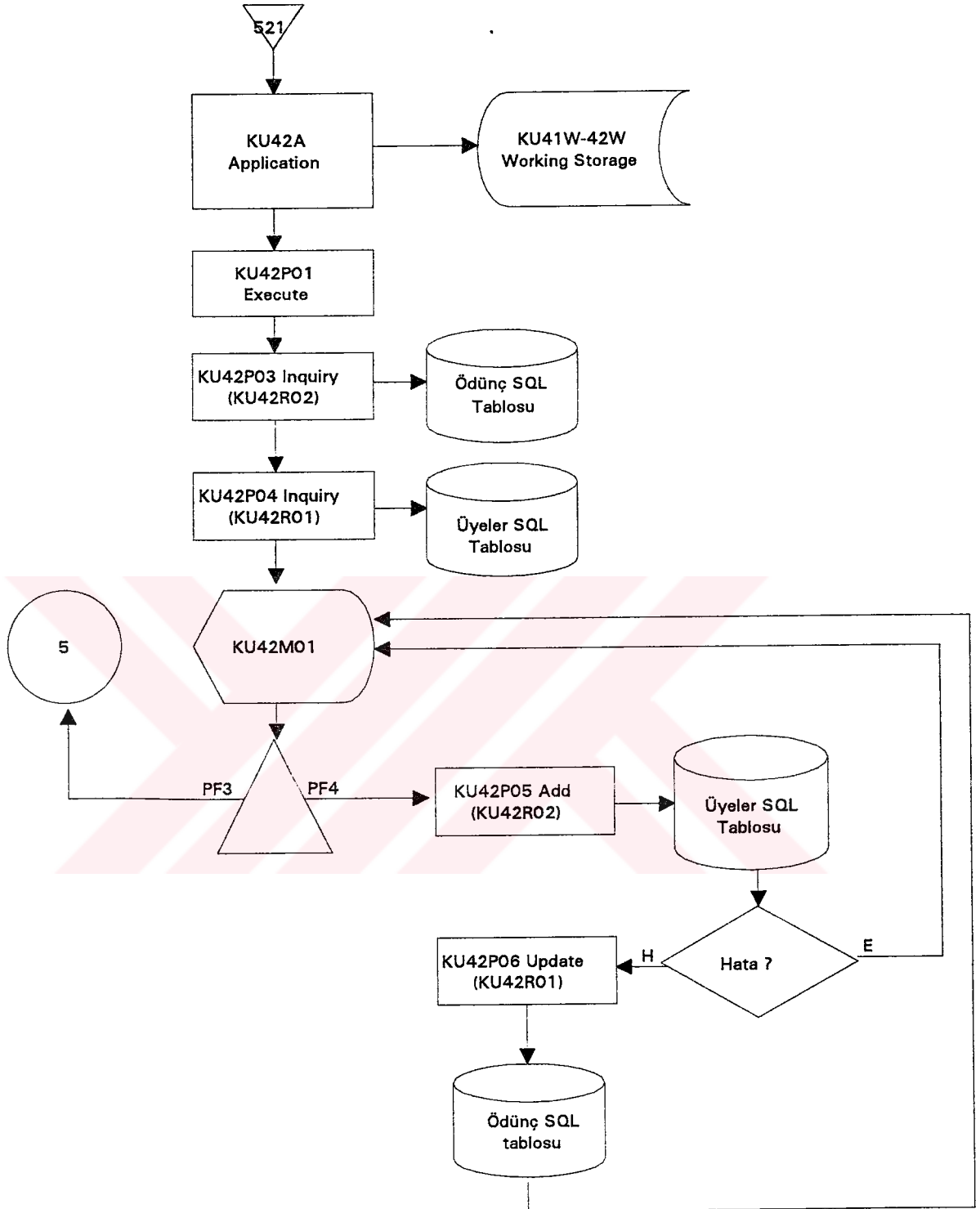
Ek 5.21. KU36A Uygulama Akış Şeması



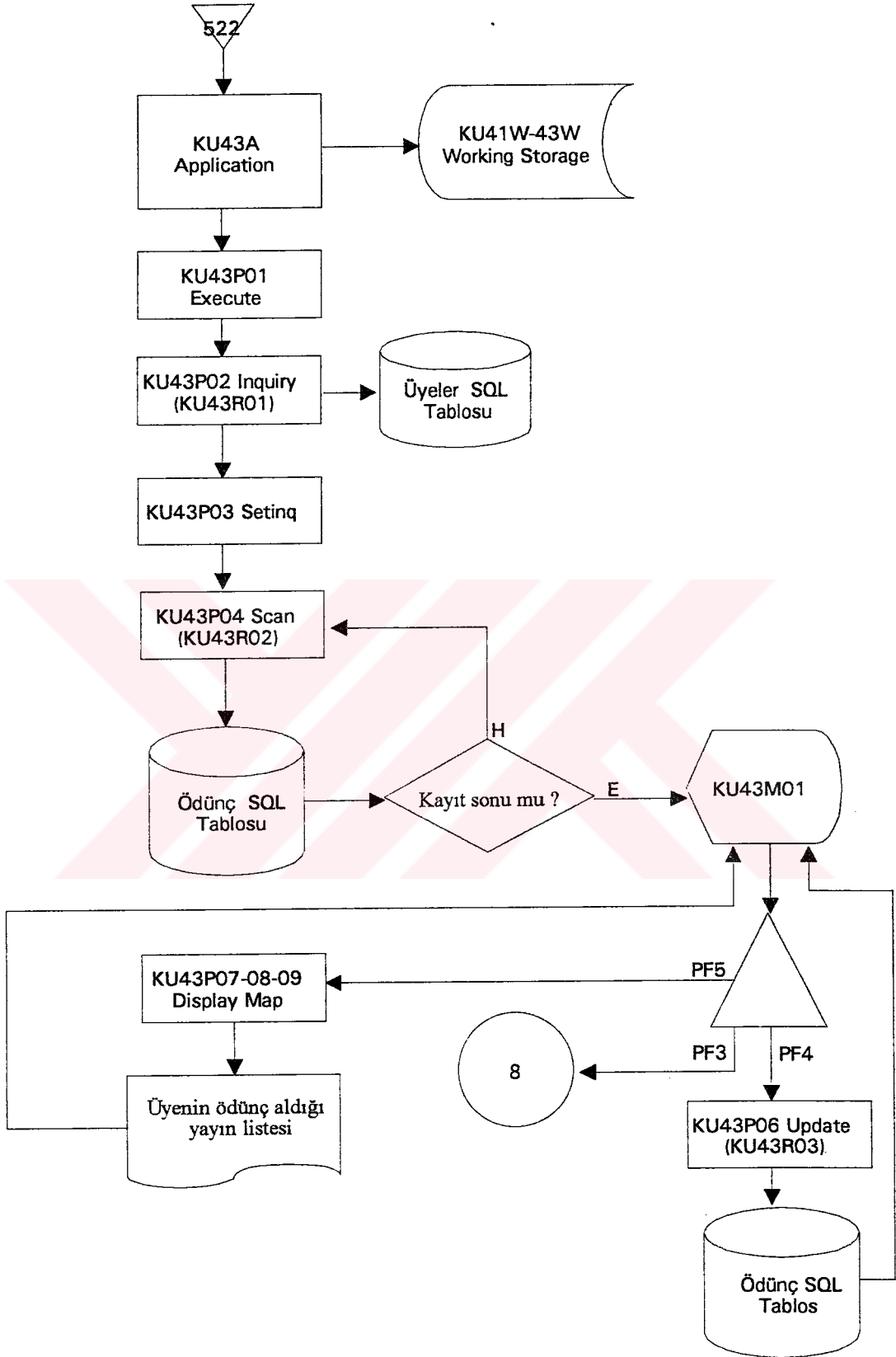
Ek 5.23. KU40A Uygulama Akış Şeması



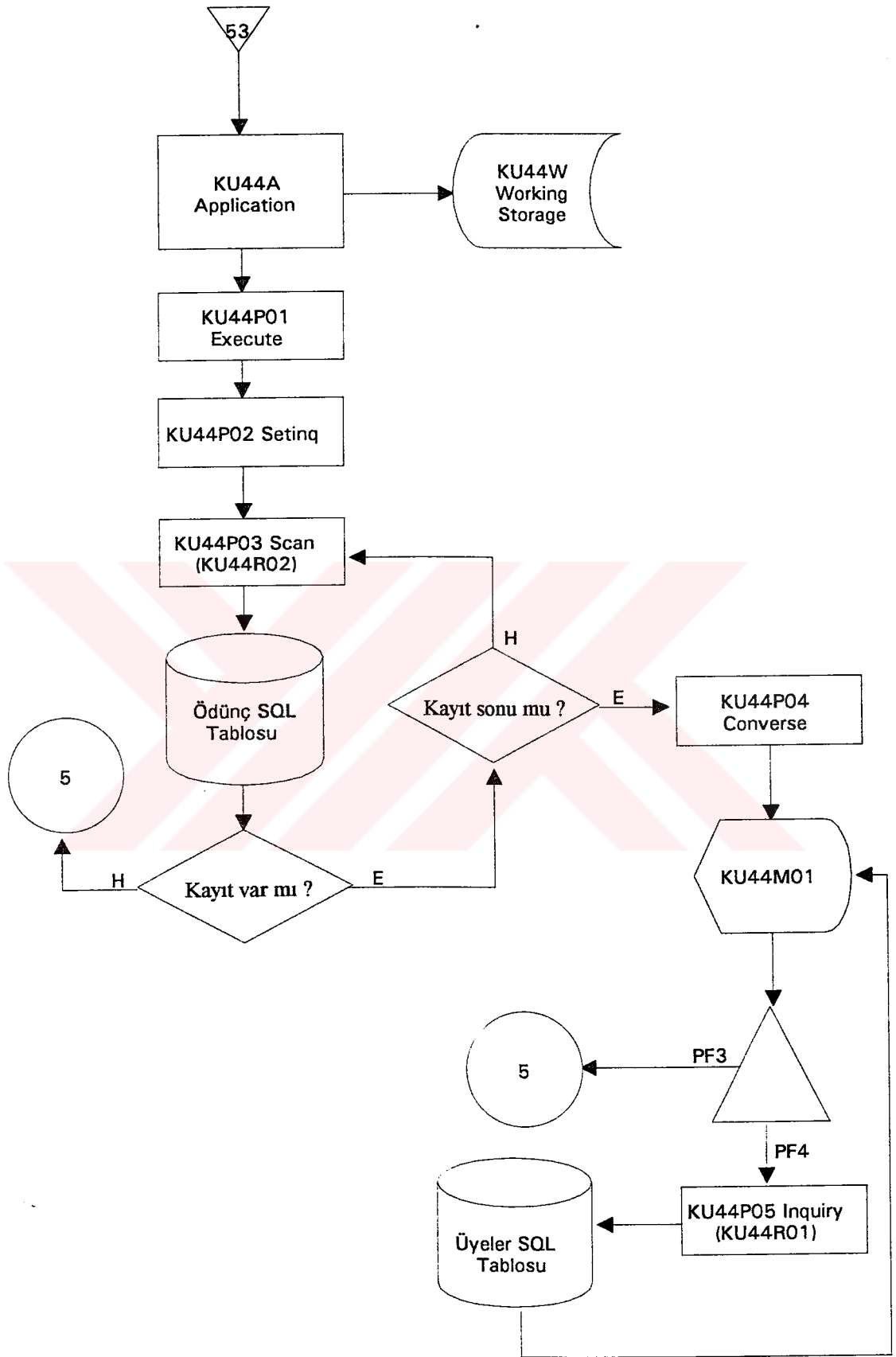
Ek 5.24. KU41A Uygulama Akış Şeması



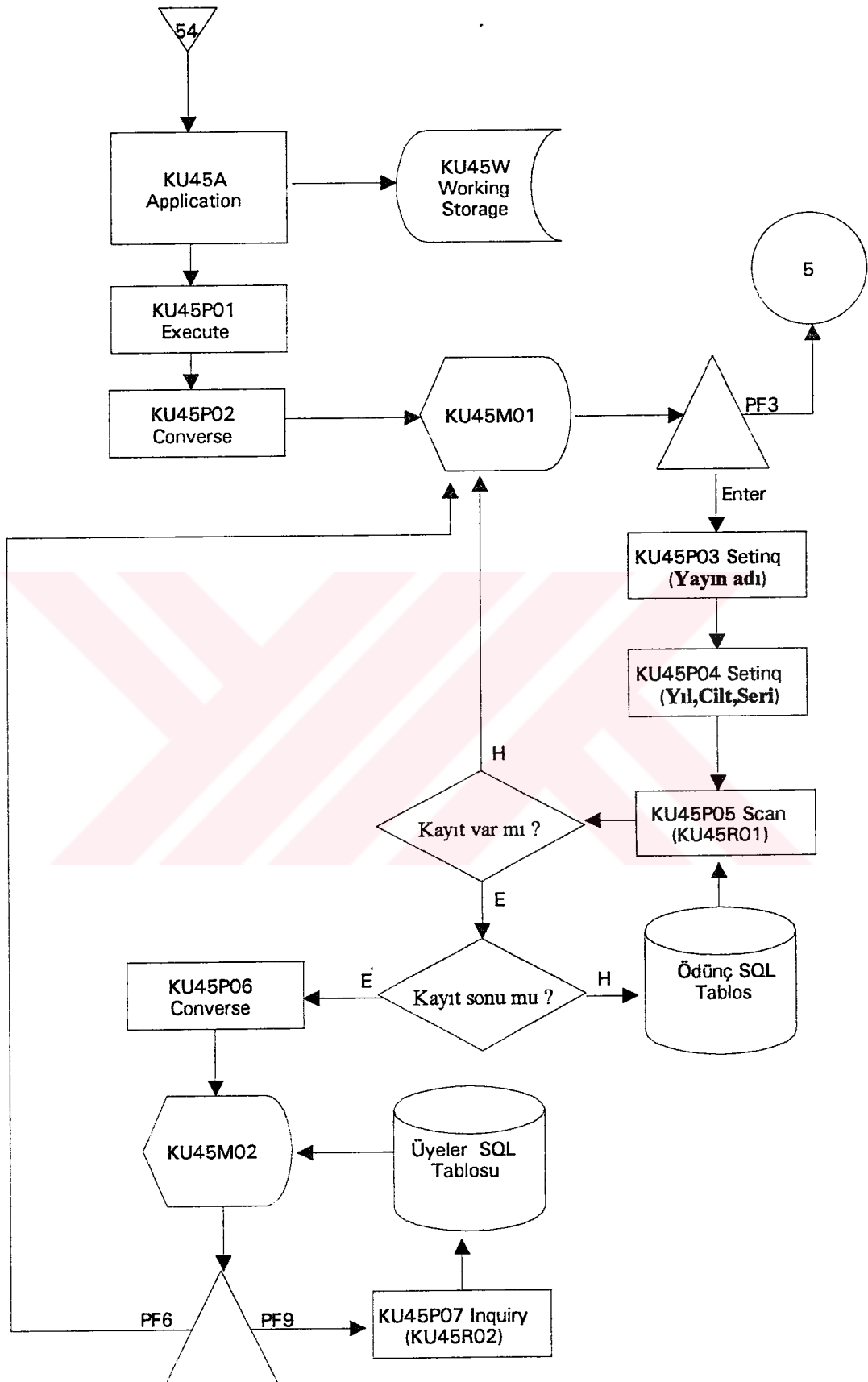
Ek 5.25. KU42A Uygulama Akış Şeması



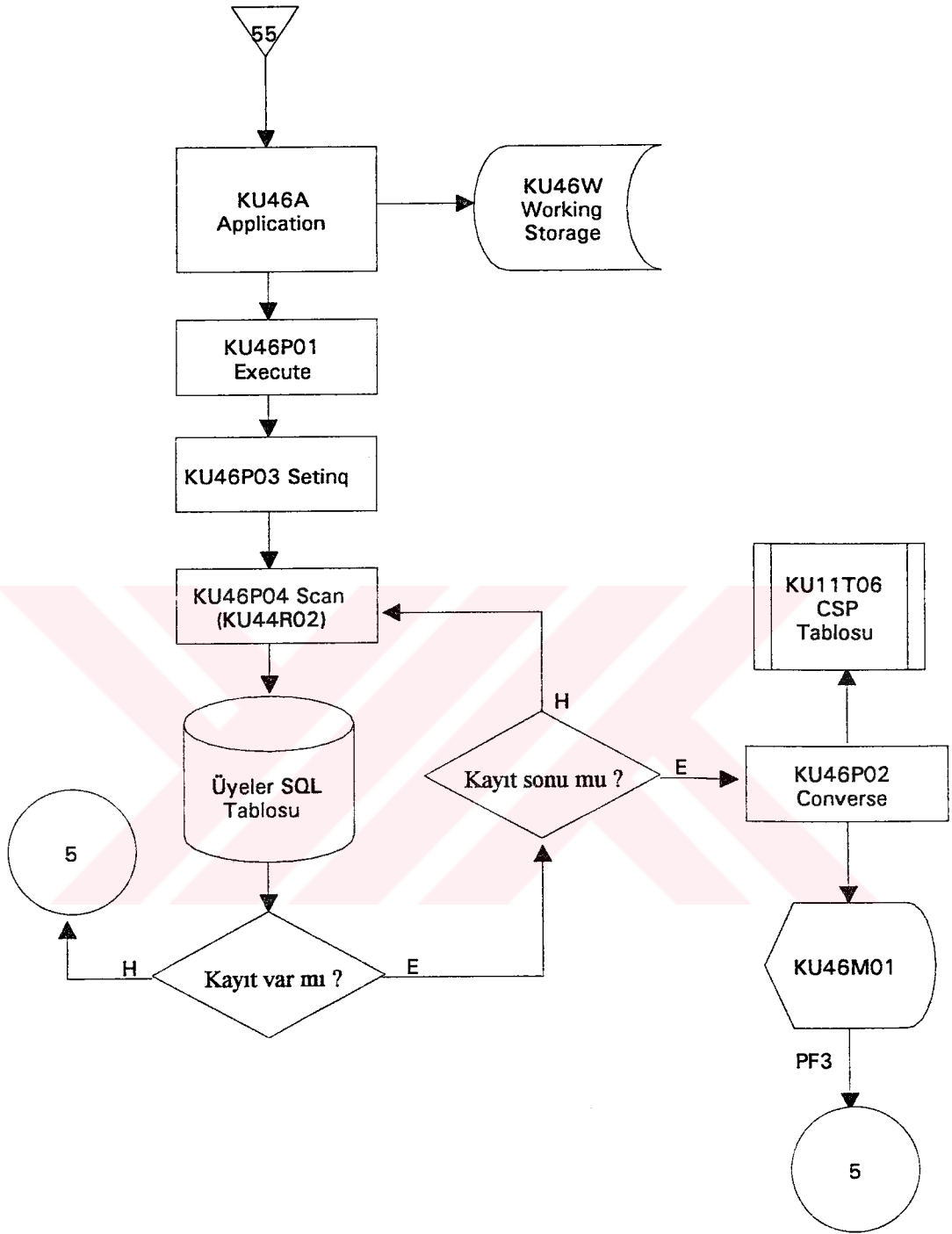
Ek 5.26. KU43A Uygulama Akış Şeması



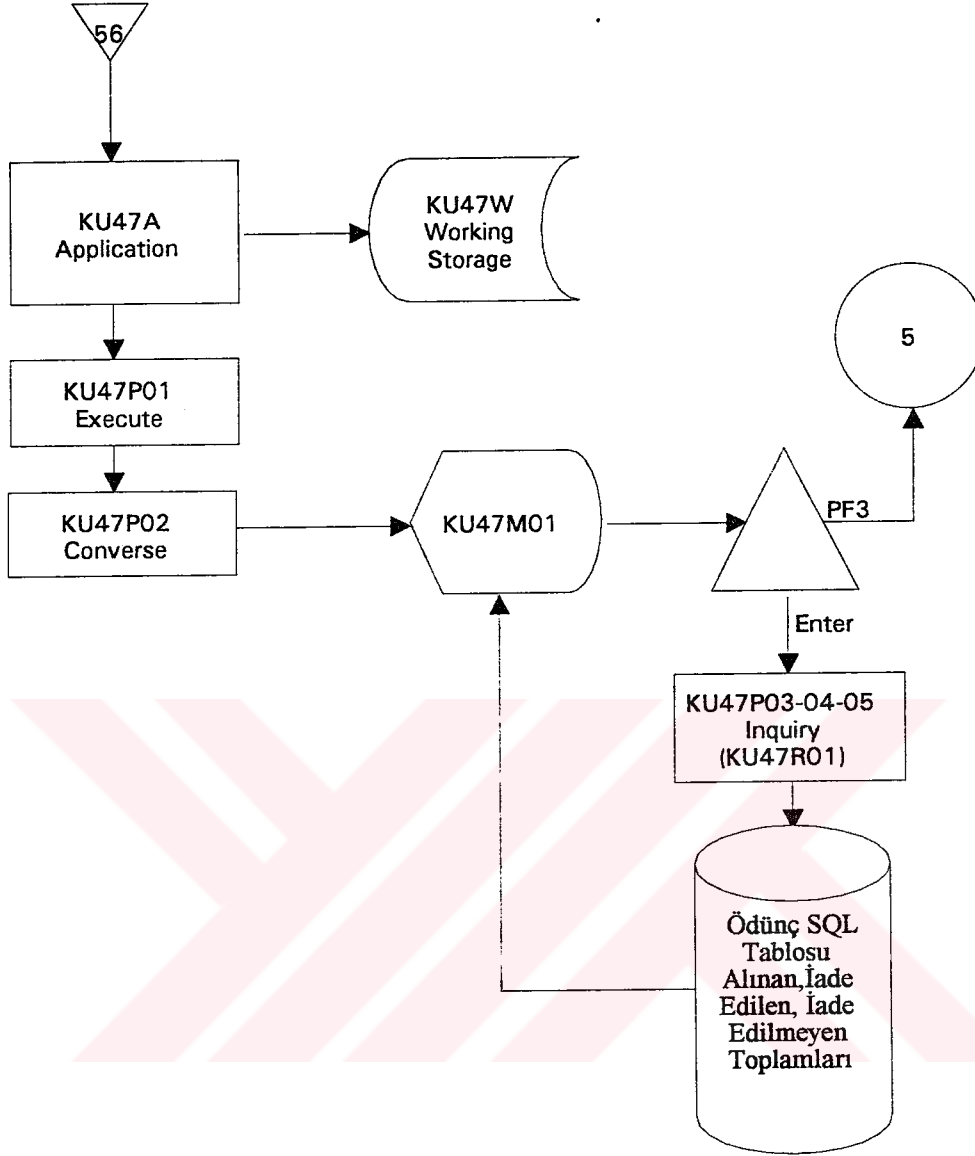
Ek 5.27. KU44A Uygulama Akış Şeması



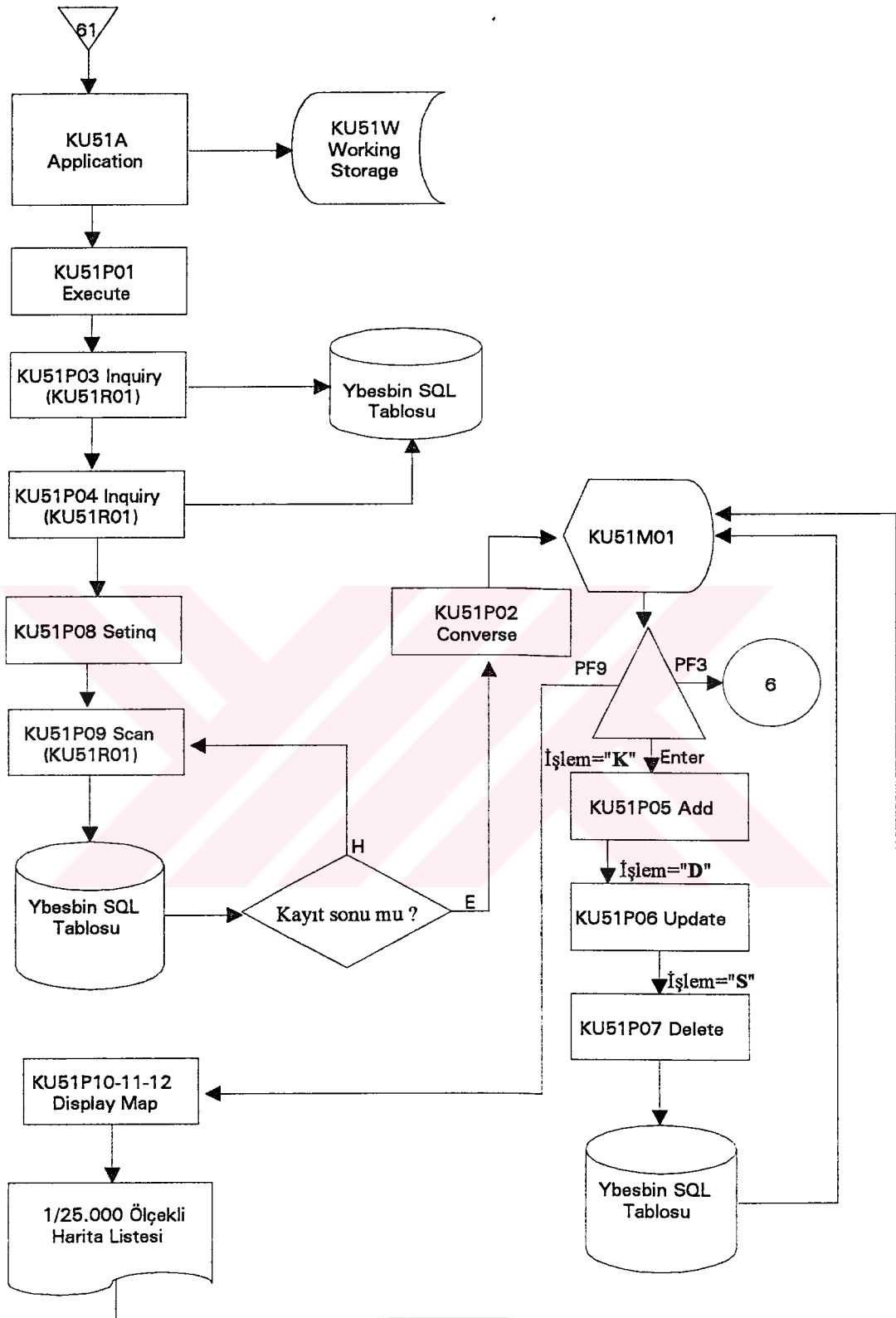
Ek 5.28. KU45A Uygulama Akış Şeması



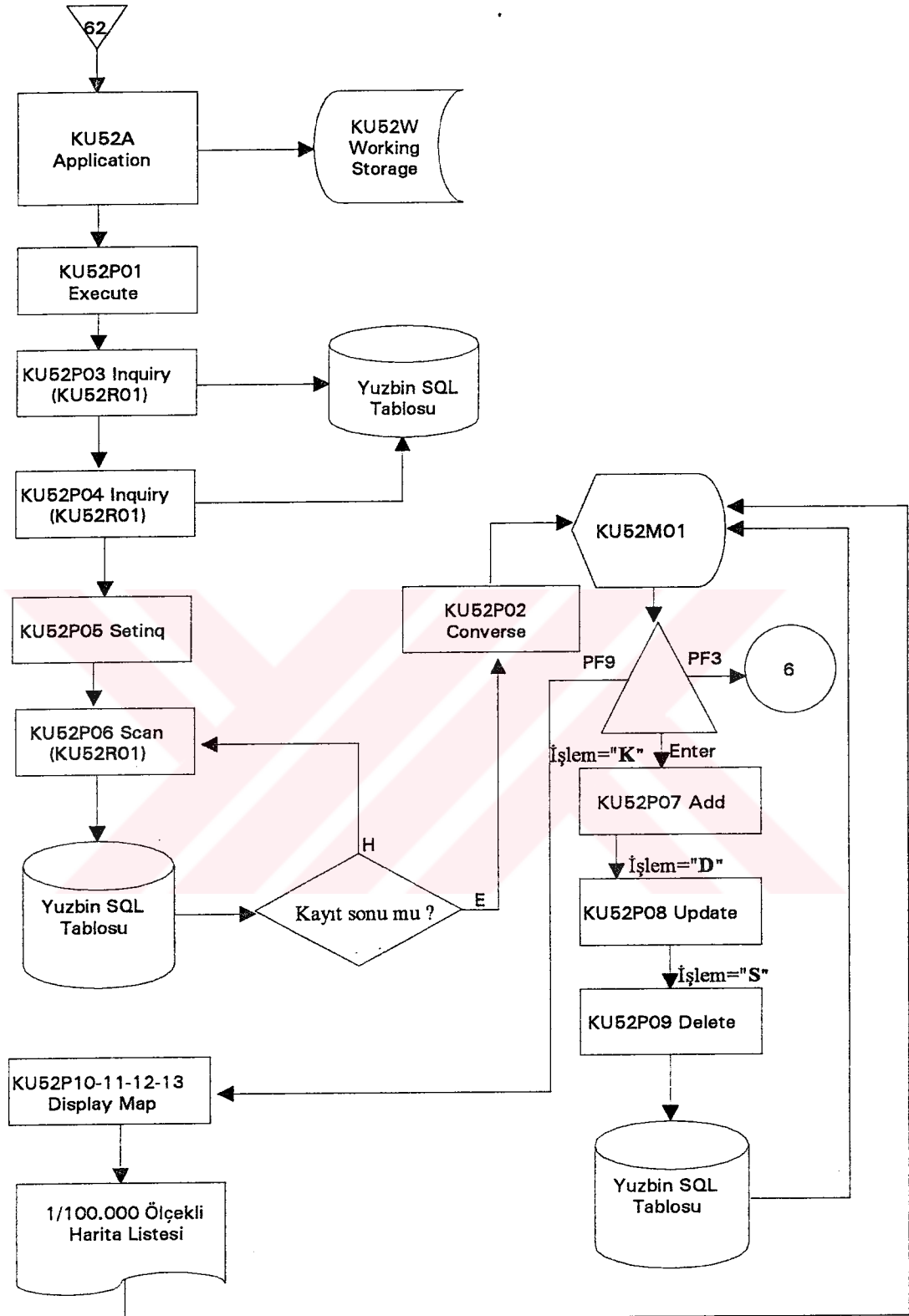
Ek 5.29. KU46A Uygulama Akış Şeması



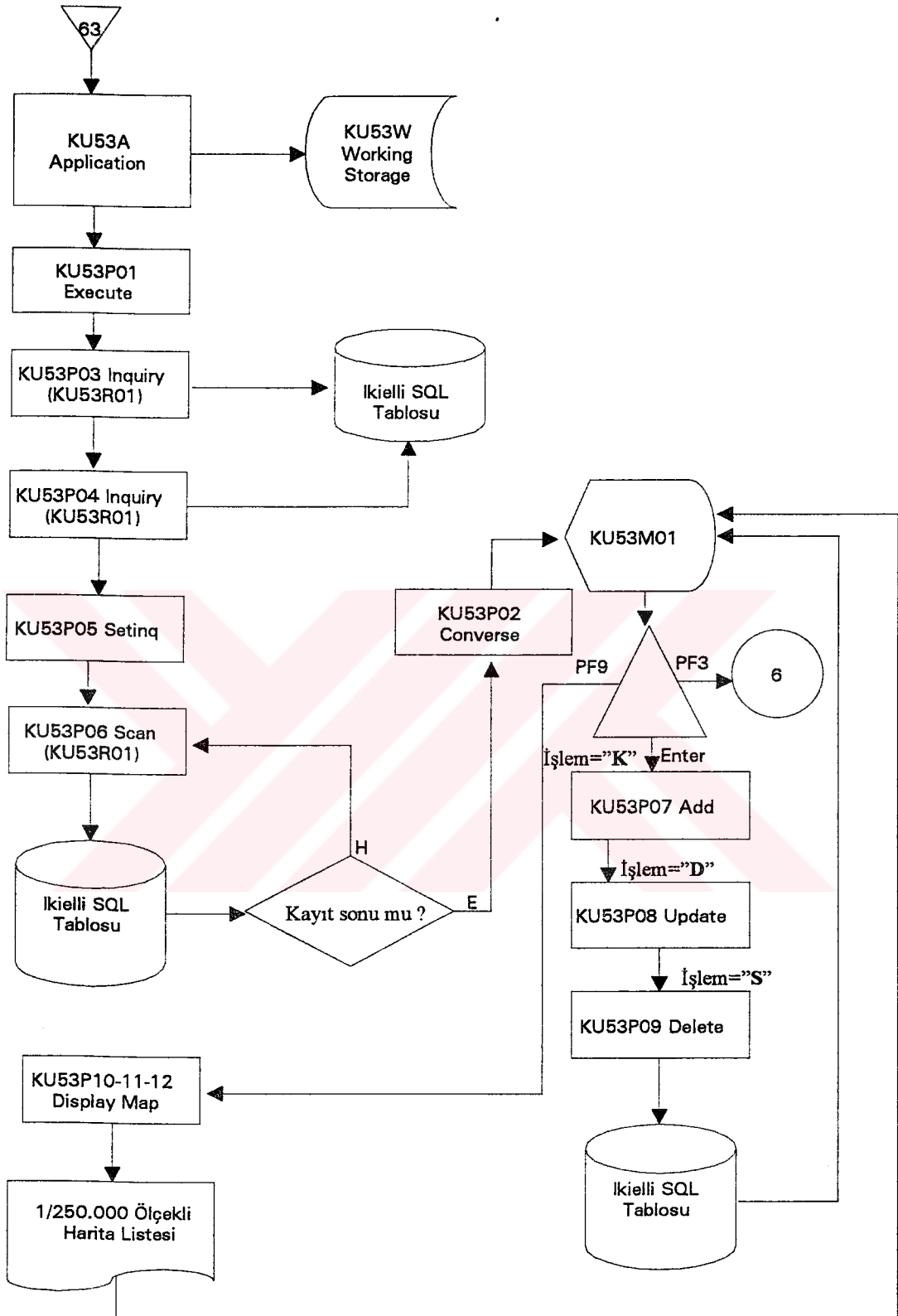
Ek 5.30. KU47A Uygulama Akış Şeması



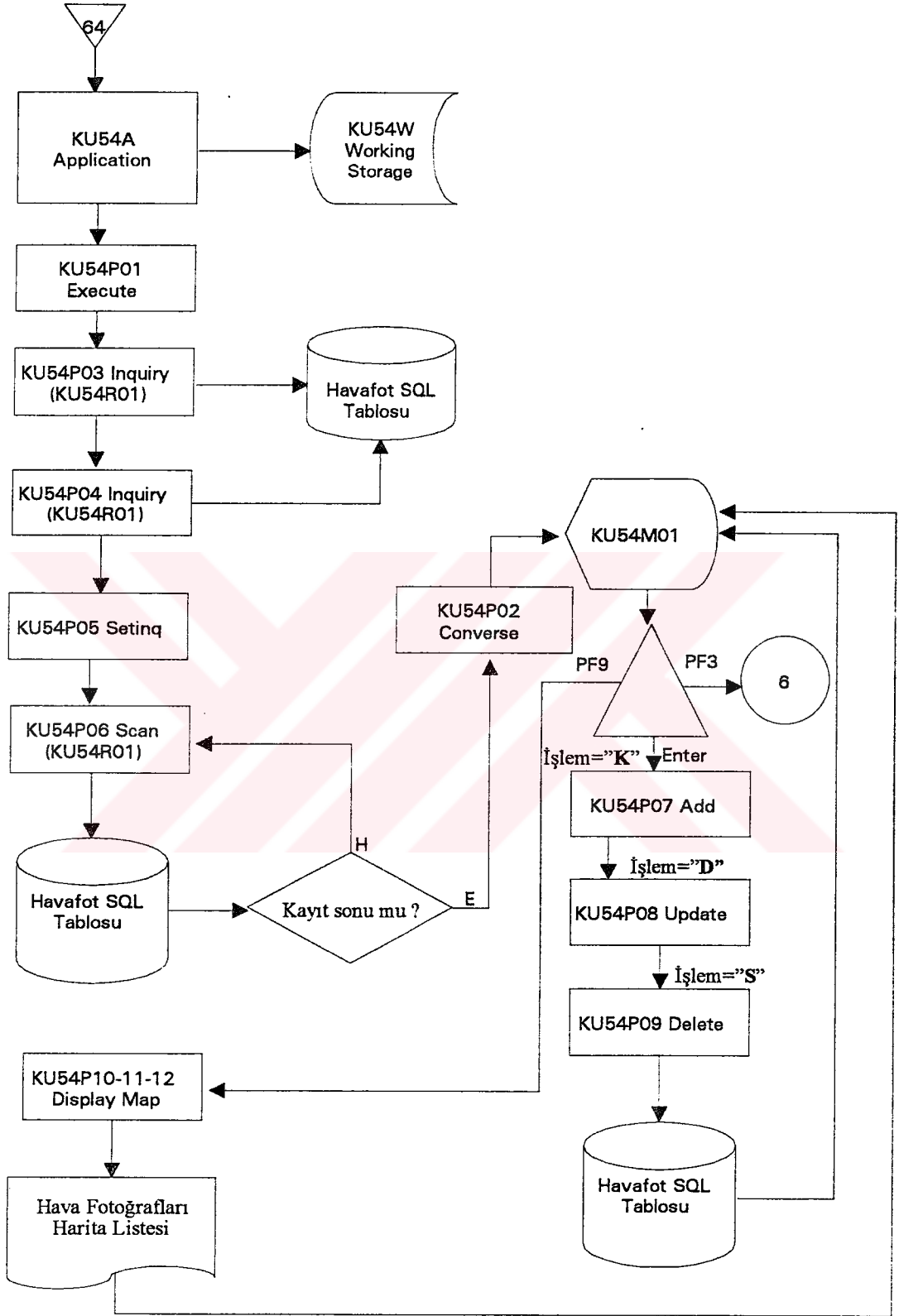
Ek 5.31. KU51A Uygulama Akış Şeması



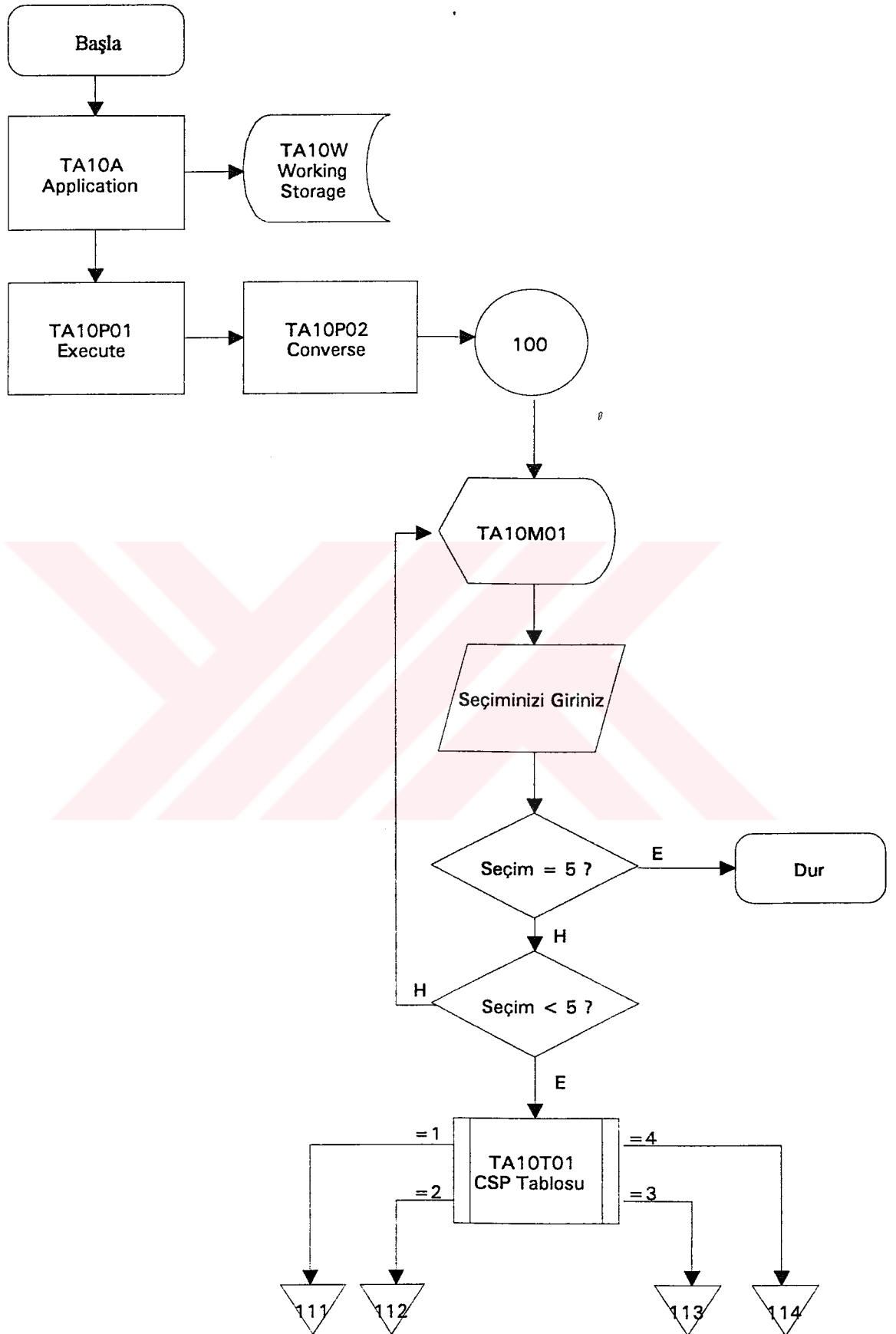
Ek 5.32. KU52A Uygulama Akış Şeması



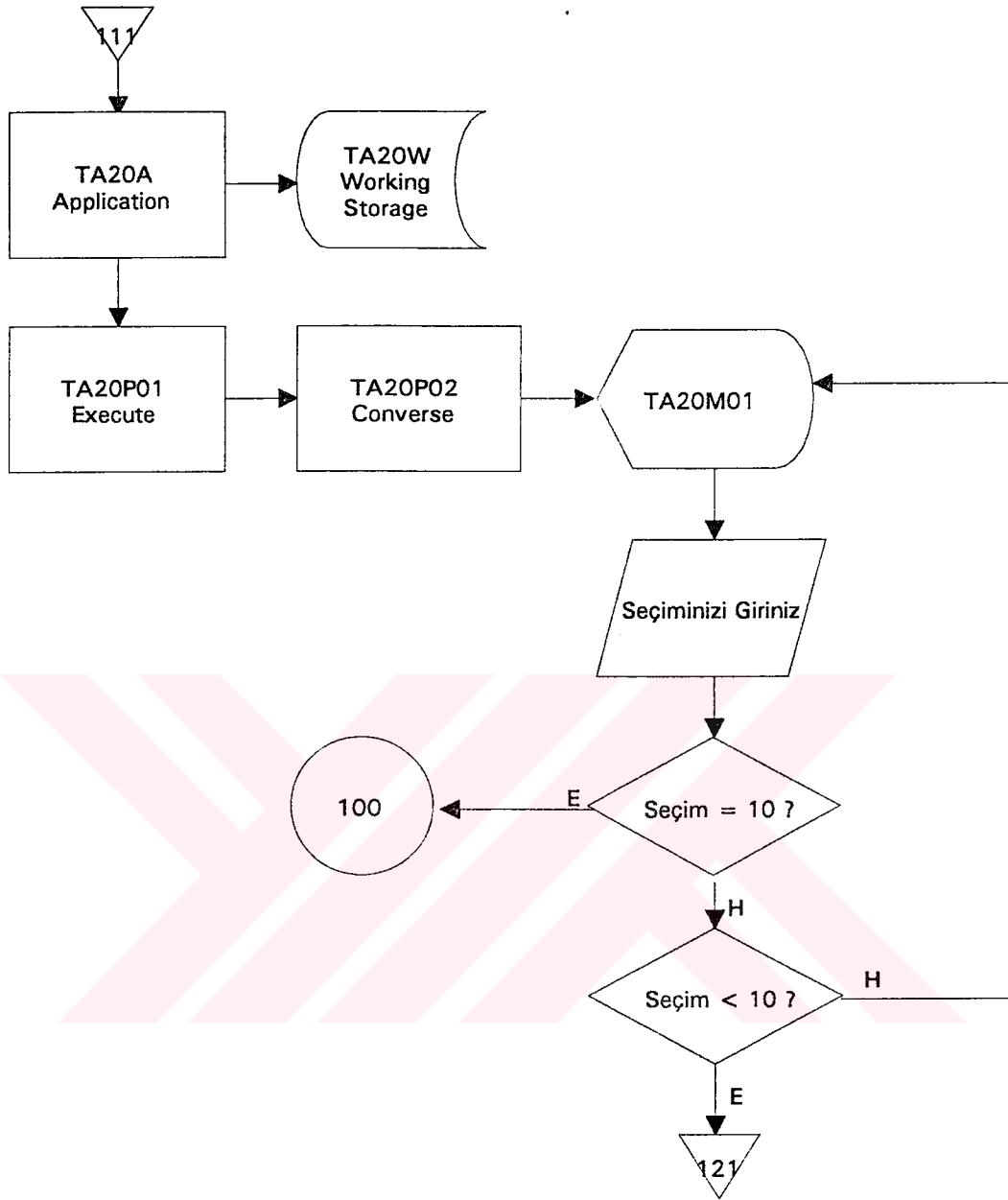
Ek 5.33. KU53A Uygulama Akış Şeması



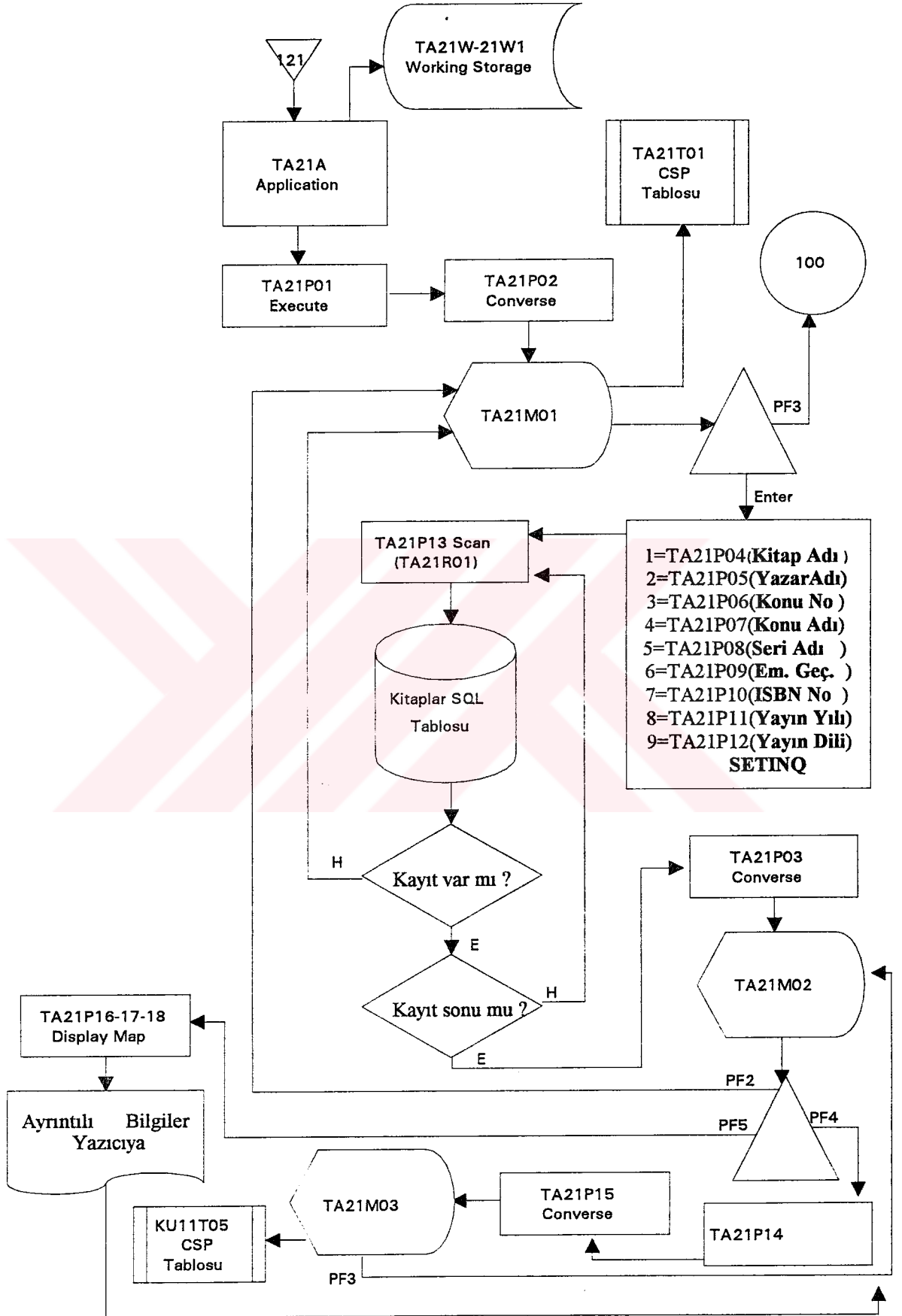
Ek 5.34. KU54A Uygulama Akış Şeması



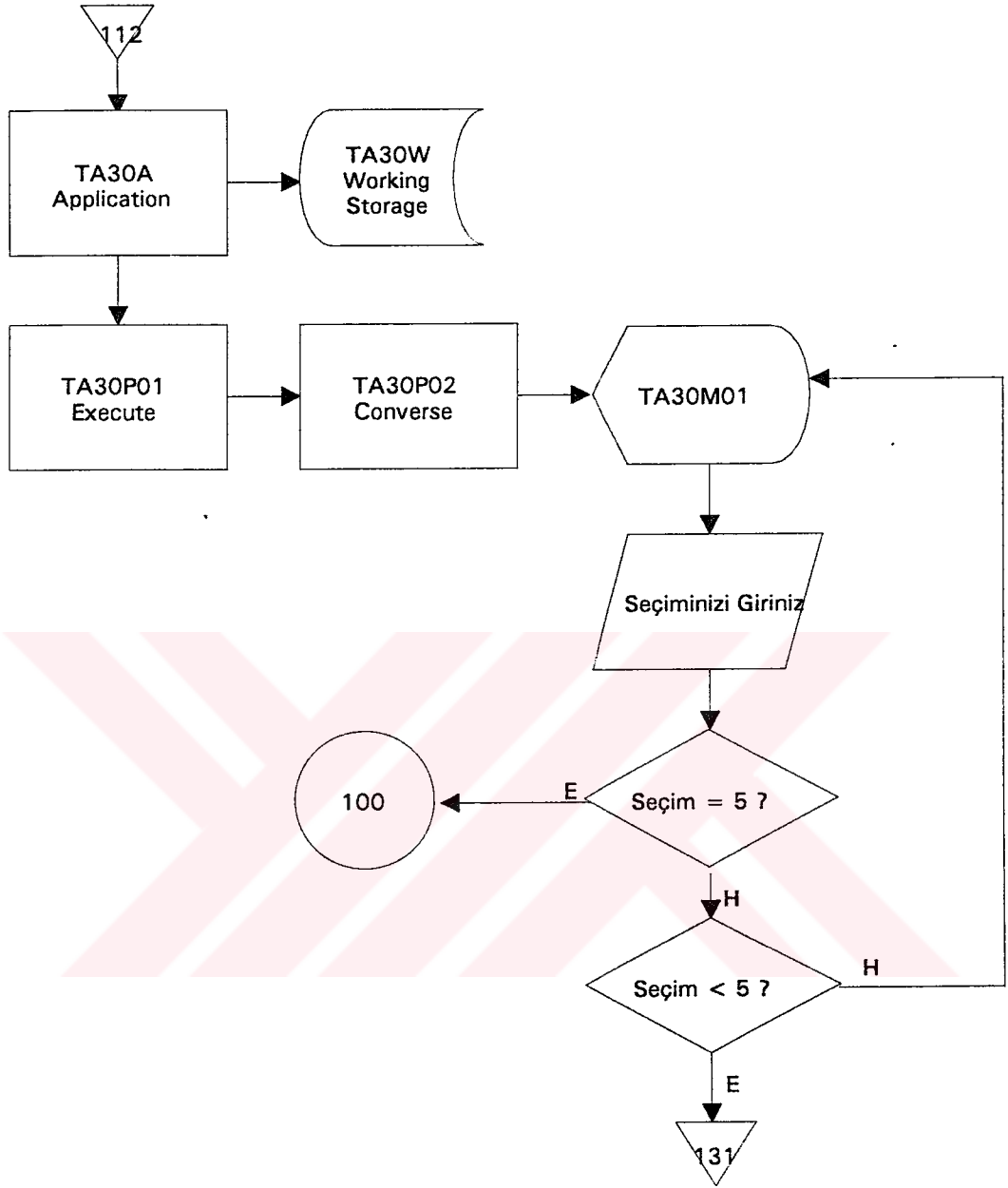
Ek 5.35. TA10A Uygulama Akış Şeması



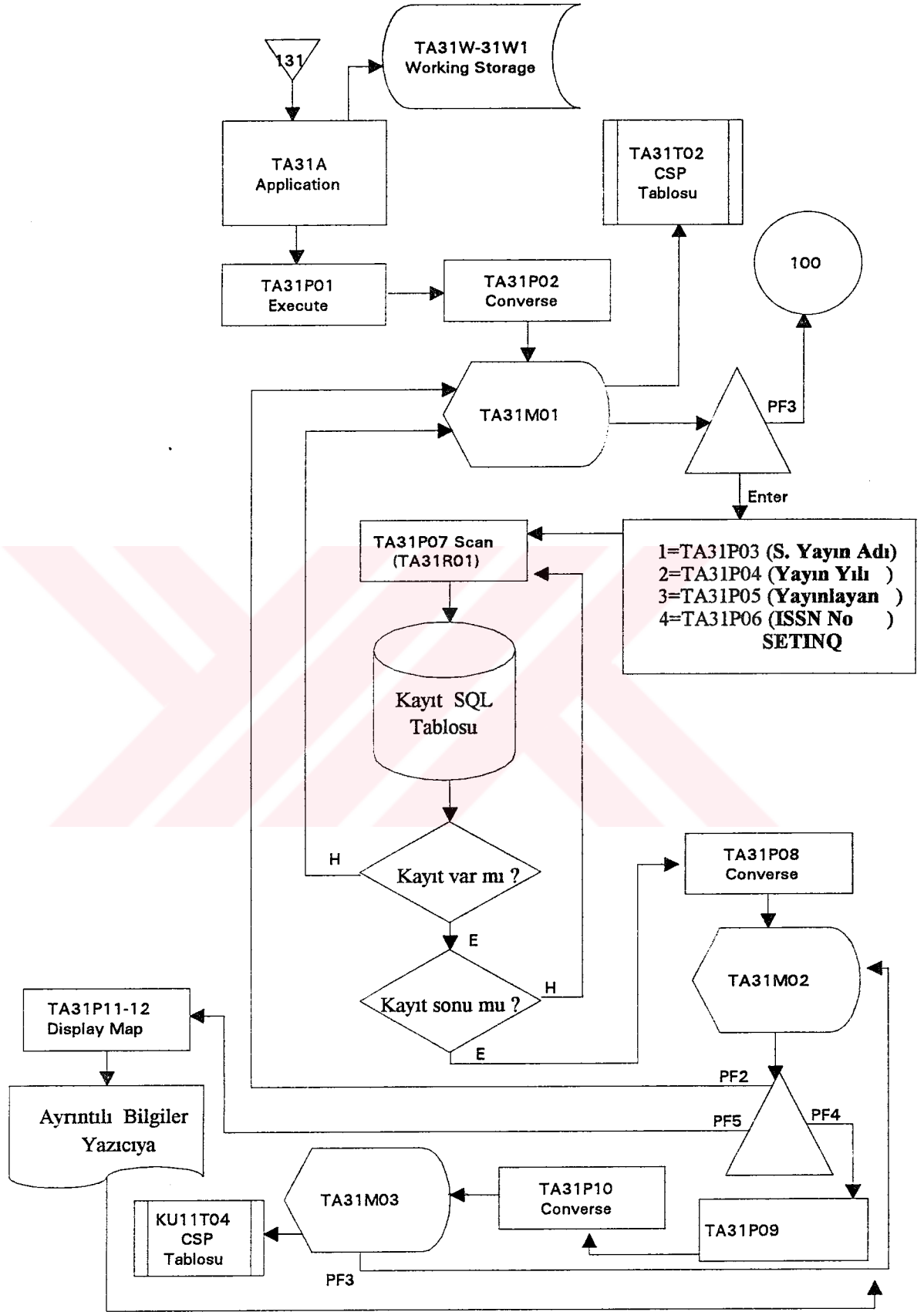
Ek 5.36. TA20A Uygulama Akış Şeması



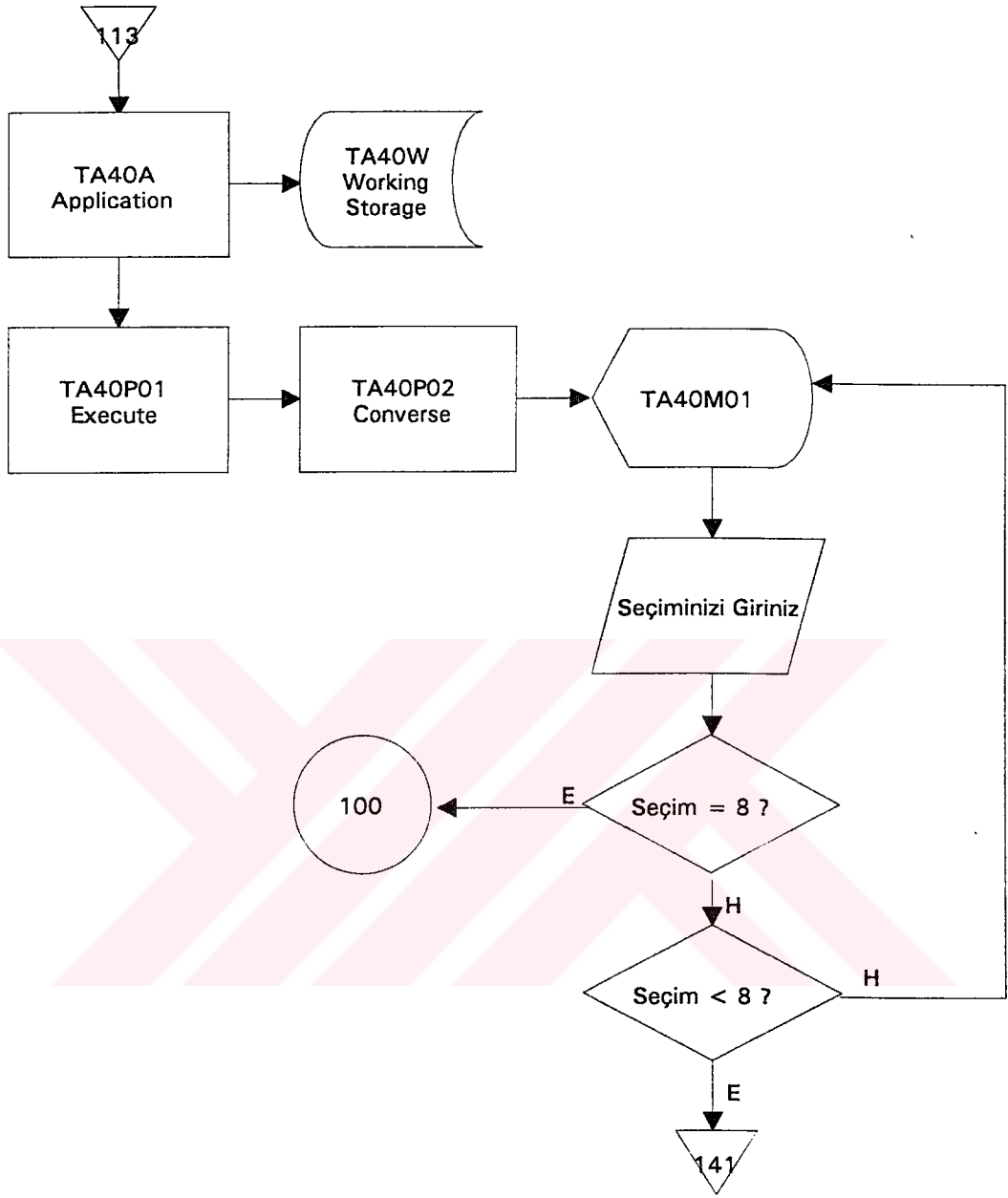
Ek 5.37. TA21A Uygulama Akış Şeması



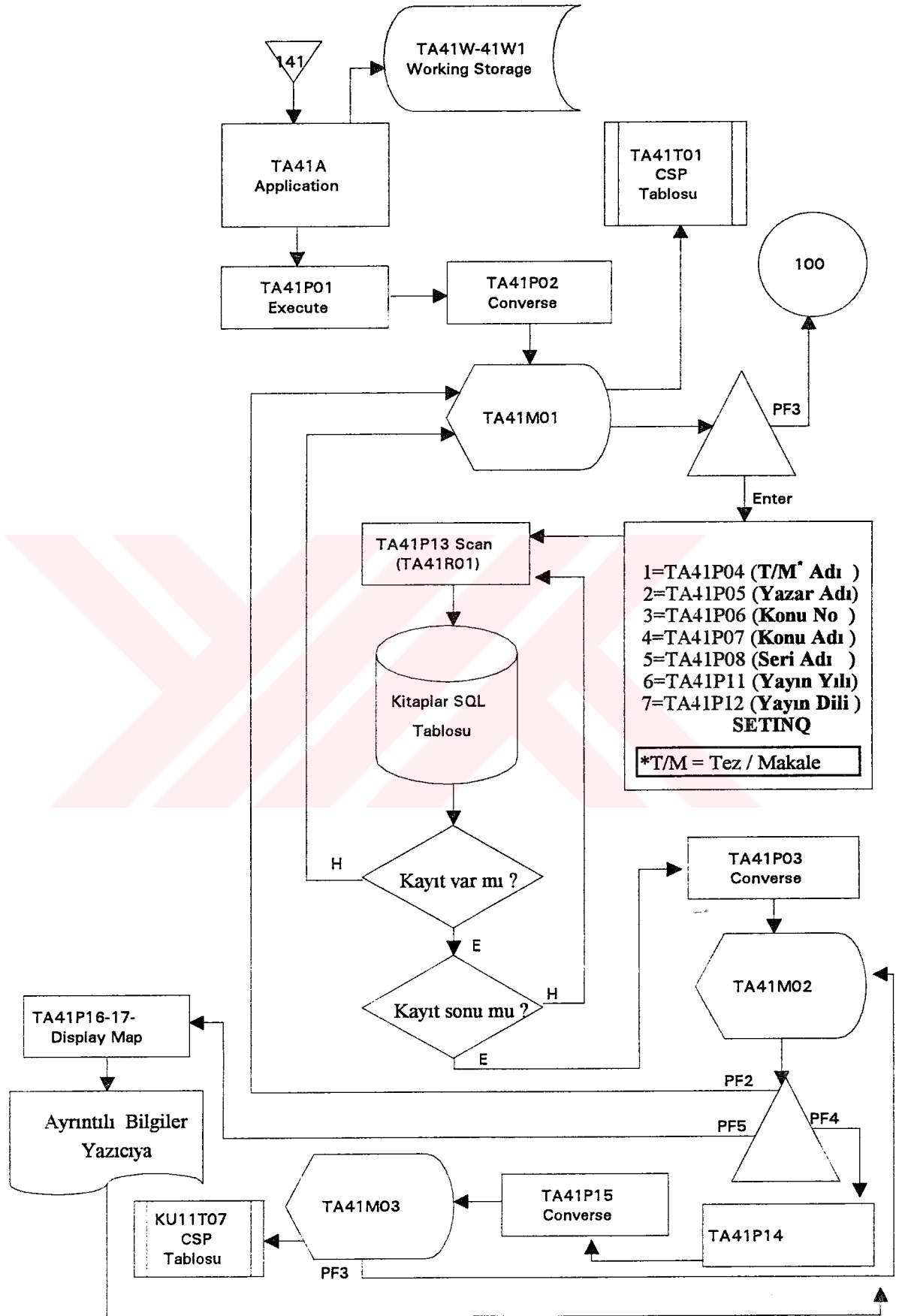
Ek 5.38. TA30A Uygulama Akış Şeması



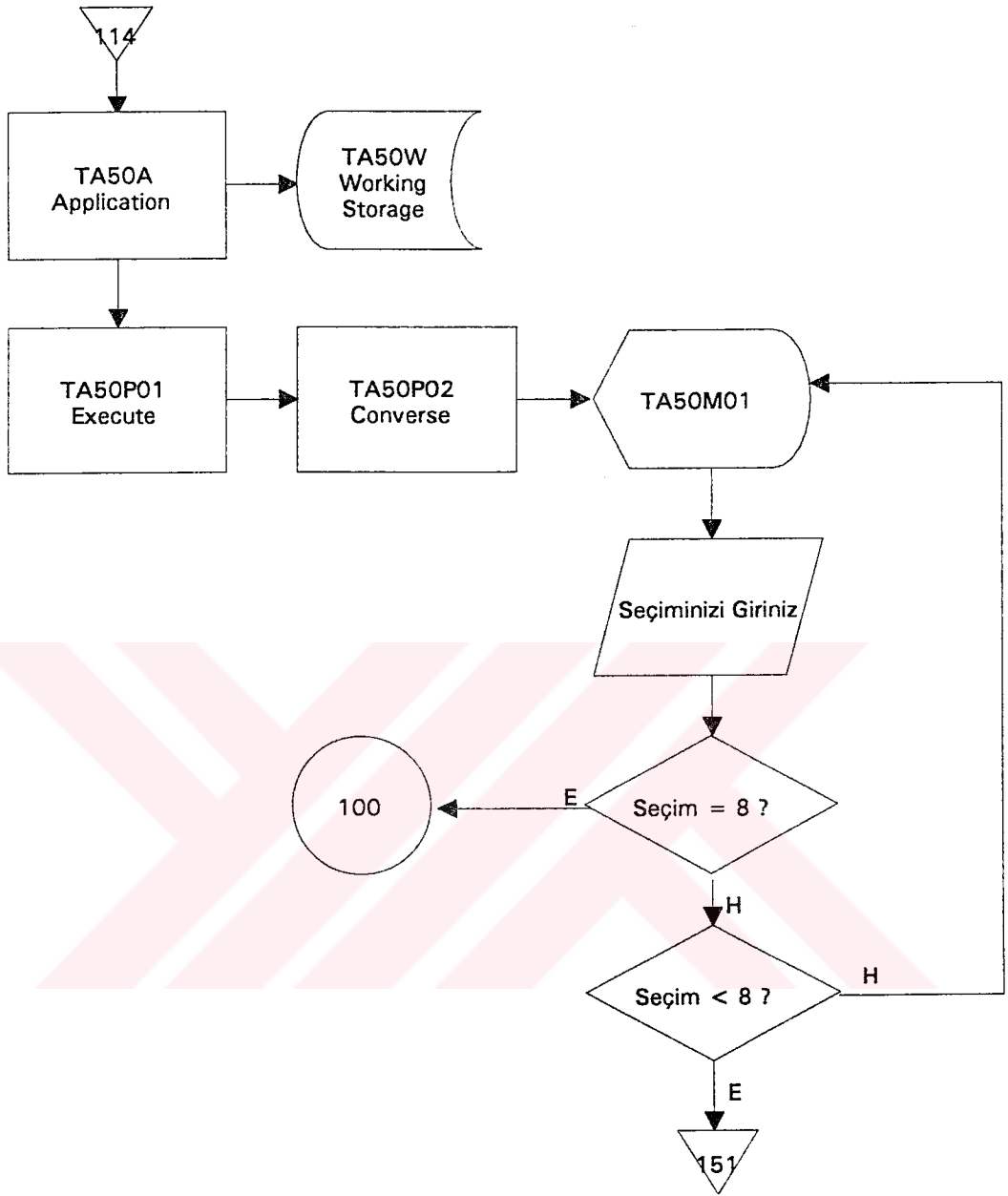
Ek 5.39. TA31A Uygulama Akış Şeması



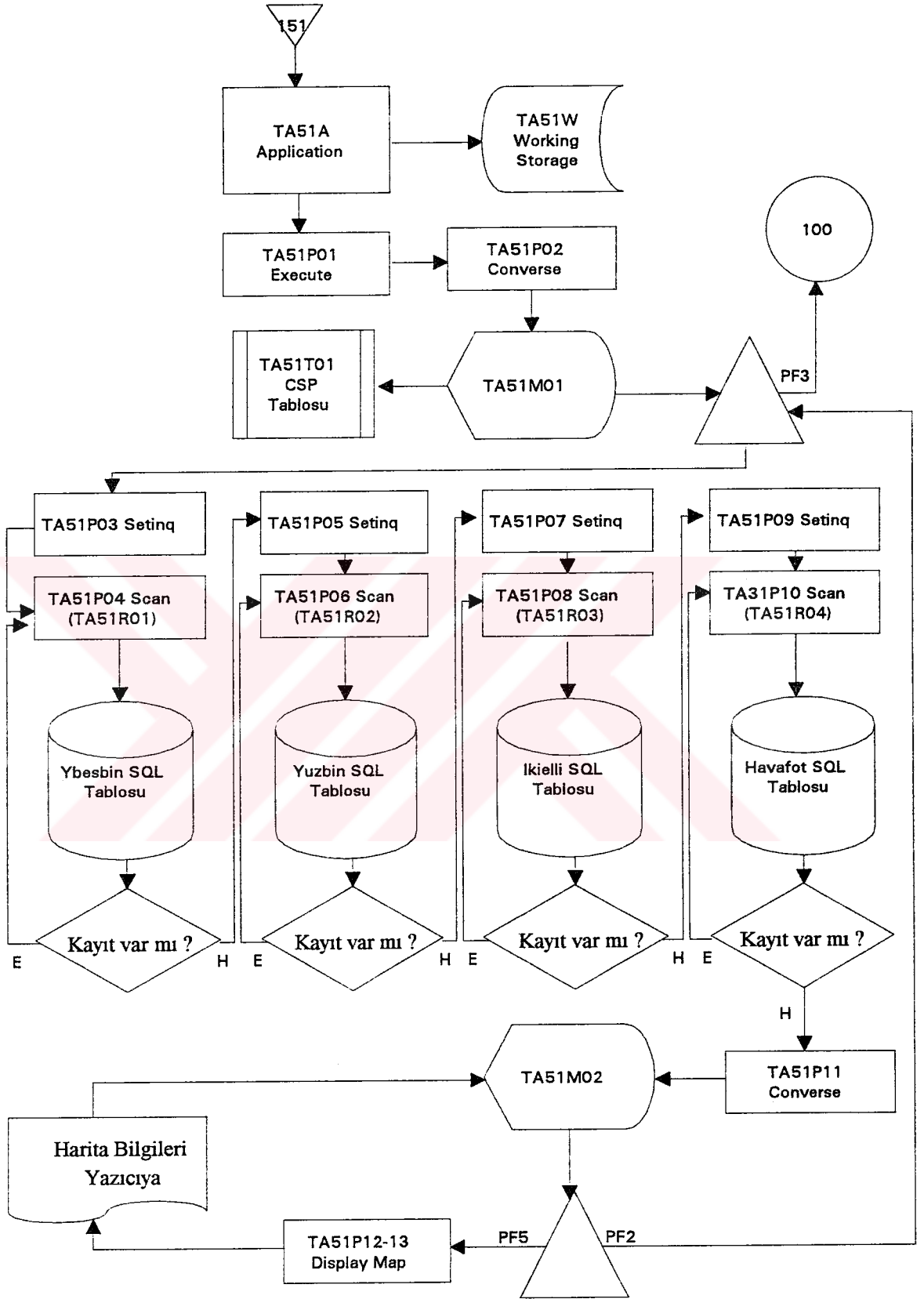
Ek 5.40. TA40A Uygulama Akış Şeması



Ek 5.41. TA41A Uygulama Akış Şeması



Ek 5.42. TA50A Uygulama Akış Şeması



Ek 5.43. TA51A Uygulama Akış Şeması

Ek 6. Program Çıktıları

Sayfa :

ISSN NUMBER	TITLE
0149-1423	AAPG BULLETIN-AMERICAN ASSO. OF PETROLEUMGEOLIST
0044-586X	ACAROLOGIA
0001-4826	ACCOUNTING REVIEW
0889-325X	ACI MATERIALS JOURNAL - AMERICAN COCRETE INSTITUE
0001-5180	ACTA ANATOMICA
0904-213X	ACTA CHEMICA SCANDINAVICA
0108-7673	ACTA CRYSTALLOGRAPHICA SECTION-A
0108-7681	ACTA CRYSTALLOGRAPHICA SECTION-B
0001-5547	ACTA CYTOLOGICA

Ek 6.1. Süreli yayın adına göre çıktı örneği

Firma Adı : Blackwell's

Sayfa :

ISSN NUMBER	TITLE
0044-586X	ACAROLOGIA
0904-213X	ACTA CHEMICA SCANDINAVICA
0108-7673	ACTA CRYSTALLOGRAPHICA SECTION-A
0108-7681	ACTA CRYSTALLOGRAPHICA SECTION-B
0001-5547	ACTA CYTOLOGICA
0300-8835	ACTA OBSTETRICA ET GYNECOLOGICA SCANDINAVICA-SUP.
0001-6446	ACTA ORIENTALIA
0001-6489	ACTA OTO-LARYNGOLOGICA-SUP.

Ek 6.2. Süreli yayın firma adına göre çıktı örneği

19__ / 19__ Arası Faturaların Parasal Dağılımı

Para Birimi	Sale Invoice	Extra Charge	Credit Note
USD			
DM			
UKS			
TL			
HF			
JY			
Toplam			

Ek 6.3. Faturaların parasal dağılımı çıktı örneği

Firma Adı : _____
 Fatura No : _____ ile ödenen yayın listesidir.

Sayfa :

ISSN Numarası	Yayının Adı	Volüm	Fatura Tutarı	Para Birimi
Sale Invoice Extra Charge Credit Note				

Ek 6.4. Belli bir fatura ile ödenen yayınların listesi çıktı örneği

Sağlama Servisi Tüm Siparişlerin Listesi

Sayfa :

Sipariş No	ISBN-ISSN No	Yayının Adı
1	0001-5547	ACTA CYTOLOGICA
2	0001-6446	ACTA ORIENTALIA
3	0001-7000	ACTA STOMOLOGICA BELGICA

Ek 6.5. Verilen siparişlerin listesi çıktı örneği

Yayın Adı : _____
 Firma Adı : _____
 ISSN No : _____

Sayfa :

Yılı	Fatura No	Fatura Tarihi	Miktarı	Para Birimi	Volüm-Cilt	Fat. Türü
Toplam						

Ek 6.6. Muhasebe işlemleri listesi çıktı örneği

ÖDÜNÇ VERİLEN YAYIN LİSTESİ

Kart No : Ünvani :
 Sicil No : Kayıt Tarihi :
 Adı : Telefon :
 Soyadı :
 Bölümü :

Verilen Yayın Adı	Alış Tarihi	Yılı	Cilt	Verilen Sayılar

Ek 6.7. Ödünç verilen yayınlar çıktı örneği

Haritanın Yer Adı	A				B				C				D				Adedi
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Toplam																	

Ek 6.8. 1/25.000 ölçekli harita çıktısı

Sıra No	İl Adı	Pafta No	İl Adı	Pafta No	Adedi
Toplam					

Ek 6.9. 1/100.000 ölçekli harita çıktısı

Sıra No	İl Adı	Plaka No	Adedi
Toplam			

Ek 6.10. 1/250.000 ölçekli harita çıktısı

Ebat	Özel No	Not	Adedi
Toplam			

Ek 6.11. Hava fotoğrafları çıktısı

[illegible][illegible][illegible][illegible][illegible]

621.382

Ş 34

Şeker, S. Selim

İletişim sistemlerinin planlanması .— / Ş. Selim Şeker .—
İstanbul : Boğaziçi Üniversitesi, 1993.

144 s. ; 24 sm .— (Boğaziçi Üniversitesi yayın no : 525)

ISBN : 975-518-026-5

I- K.a. II- Konu. III- Seri. I- Telekomünikasyon
sistemleri — Planlama

Ek 6.14. Yazar adına göre çıkarılan katalog kartı

621.382

Ş 34

İletişim sistemlerinin planlanması

Şeker, S. Selim

İletişim sistemlerinin planlanması .— / Ş. Selim Şeker .—
İstanbul : Boğaziçi Üniversitesi, 1993.

144 s. ; 24 sm .— (Boğaziçi Üniversitesi yayın no : 525)

ISBN : 975-518-026-5

Ek 6.15. Kitap adına göre çıkarılan katalog kartı

621.382

Ş 34

TELEKOMÜNİKASYON SİSTEMLERİ — PLANLAMA

Şeker, S. Selim

İletişim sistemlerinin planlanması .— / Ş. Selim Şeker .—
İstanbul : Boğaziçi Üniversitesi, 1993.

144 s. ; 24 sm .— (Boğaziçi Üniversitesi yayın no : 525)

ISBN : 975-518-026-5

Ek 6.16. Konu adına göre çıkarılan katalog kartı

621.382

Ş 34

(Boğaziçi Üniversitesi yayın no : 525)

Şeker, S. Selim

İletişim sistemlerinin planlanması .— / Ş. Selim Şeker .—
İstanbul : Boğaziçi Üniversitesi, 1993.

144 s. ; 24 sm .— (Boğaziçi Üniversitesi yayın no : 525)

ISBN : 975-518-026-5

Ek 6.17. Seri adına göre çıkarılan katalog kartı

621.382

69023

Ş 34

Şeker, S. Selim

İletişim sistemlerinin planlanması .— / Ş. Selim Şeker .—
İstanbul : Boğaziçi Üniversitesi, 1993.

144 s. ; 24 sm .— (Boğaziçi Üniversitesi yayın no : 525)

ISBN : 975-518-026-5

1 adet

Ek 6.18. Yer numarasına göre çıkarılan katalog kartı

SÜRELİ YAYINLAR KATALOGU

-A-

ACTA CYTOLOGICA

?, ?, 12

1990 , (1991) , 1992 , (1993) , 1994 , (1995) ,

ACTA ORIENTALIA

Budapest , 1950 , 6

(1989) , 1990 - 1991 , (1992) , 1993 , (1994) ,

..

...

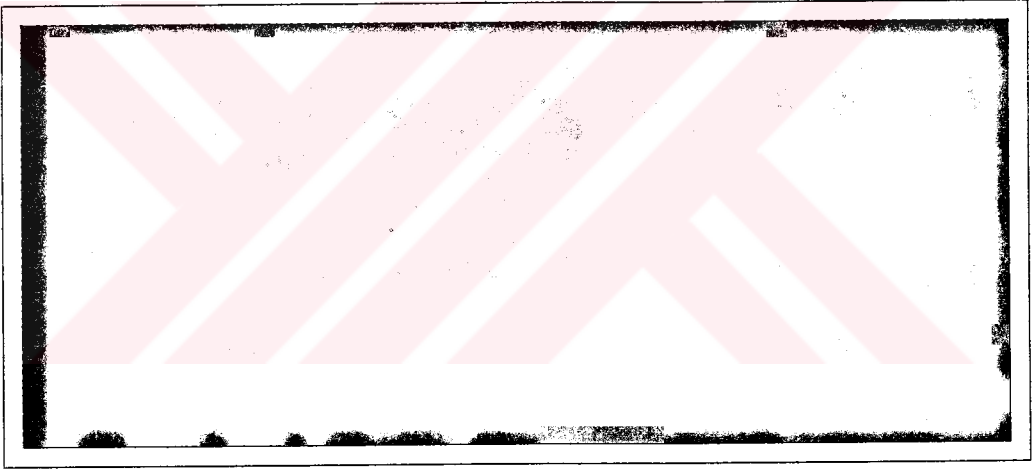
.....

Ek 6.19. Süreli yayınlar katalogu çıktısı (Alfabetik)

<u>Okuyucunun :</u>		Tarih :/...../19...
Adı :		
Soyadı :		İmza :
Kart No:		

<u>Sıra No</u>	<u>Yayının Adı</u>	<u>Yer Numarası</u>
1-		
2-		
3-		

Ek 6.20. Tasarımı düşünülen okuyucu yayın ödünç alma formu (Ön yüz)



Ek 6.21. Tasarımı düşünülen okuyucu yayın ödünç alma formu (Arka yüz)

KAYNAKLAR

- Akmen, E. ve Nazlıarslan, I.**, Kasım 1989, SQL/DS, Anadolu Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi.
- Anonymous**, 1991, Bilgisayar Ansiklopedisi, Milliyet Yayınları.
- Anonymous**, Nisan 1996, CHIP Bilgisayar Kültürü Dergisi, Sayı 4.
- Anonymous**, 1989, DBASE IV, Ekonomist Yayınevi, Ankara.
- Anonymous**, Mayıs 1988, SQL/DS Tanıtma ve Uygulama Geliştirme Kurs Notları, IBM Türk Ltd. Şti. Eğ. Müd.
- Anonymous**, 1990, Cross System Product General Information, USA, GH23-0500-4, IBM.
- Anonymous**, December 1988, Managing SQL/Data System, Version 2 Release 2, USA, SH09-8069-01, IBM.
- Anonymous**, Nisan 1996, PC World Türkiye Dergisi, Windows Veritabanları, IDG/UFT Yayıncılık, Sayı 31.
- Anonymous**, 1985, Türkiye’de Kütüphanecilik Alanında Bilgisayar Uygulamaları 1. Sempozyumu Sonuç Raporu, İstanbul Üniversitesi Edebiyat Fakültesi Kütüphanecilik Bölümü, İstanbul, 1985 - AD - 4380.
- Anonymous**, 1986, Türk Dili ve Ed. Ansiklopedisi, Dergâh Yayınları, İstanbul, Cilt 6.
- Anonymous**, 1989, Cross System Product Reference, Ege Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi.
- Anonymous**, 1992, Britannica Compton’s, Ana Yayıncılık A.Ş., İstanbul, ISBN : 975-7760-00-5, 13. Cilt.
- Anonymous**, Temmuz 1993, Bilgisayar (Datamation) Dergisi, YA-SA Yayıncılık, Yıl :16, Sayı : 147.
- Atılğan, D.**, Mart 1995, Türkiye’de Kütüphaneciliğin Gelişimi, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 9, Sayı 1, ISSN : 1300-0039.
- Aydın, Emin D.**, 1990, Veritabanı (Database), İstanbul, Evrim Basım Yayım Dağıtım.
- Aydın, Emin D.**, Ekim 1991, Bilgi Bilimi İnformatik-Genel Sistemler-Sibernetik ve Kitle İletişimi, Aydın Özel Eğitim ve Yayıncılık, İstanbul.

- Ayfer, Can U.**, Ağustos 1995, Kim Korkar Hain Bilgisayardan, 3. Basım, İstanbul, Pusula Yayıncılık ve İletişim Ltd.
- Babür, Zafer**, 1995, Bilgisayarla İletişim, Sistem Yayıncılık, İstanbul, ISBN:975-7397-50-4.
- Büke, Atilla**, DBase IV El Kitabı, 1991, A&S Yayın Tanıtım, Ankara.
- Calvert, C.**, Şubat 1997, Çev. Hasan F. Keçeci ve Esra Atalay, Delphi Unleashed, Sistem Yayıncılık, İstanbul, ISBN : 975-322-010-3.
- Chandor, Anthony**, 1985, Dictionary of Computers, Penguin Group, England, London, (4rd Edition), ISBN : 0-14-051127-X.
- Date, C. J.**, November 1987, An Introduction to Database Systems, Fourth edition, Addison-Wesley Publishing Company, Volume 1, USA, ISBN : 0-201-14201-5 (v.1).
- Fitzgerald, J.**, 1993, Business Data Communications, Basic Concepts, Security and Design, John Wiley & Sons, Inc., Singapore, Fourth Edition, ISBN : 0-471-59983-2.
- Gane, G. and Sarson, T.**, 1979, Structured System Analyses : Tools and Techniques, Prentice-Hall, Inc., Englewood Cliffs, New Jersey-USA, ISBN : 0-13-854547-2, Chapter 4.
- Güder, Gazi**, 1986, Bilgi İşlem Terimleri Sözlüğü, Birsan Yayınevi, İstanbul.
- Gustavsson, S.**, Mart 1996, Çev. Neslihan Uraz, İsveç'te Konuşan Kitaplar-Görme Özürlüler İçin Gelişmekte Olan Bir Olanak, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 10, Sayı 1, ISSN : 1300-0039.
- Haynes, W. K. Et al.**, 1988, The Cross system Product Application Generator : An Evolution, IBM System Journals, Vol. 27, No 3, U.S.A., ISSN : 18-8670.
- Keseroğlu, Hasan S.**, 1988, Kataloqlama Kuralları Örnekleri ile Anglo-Amerikan Kataloqlama Kuralları, Edebiyat Fakültesi Basımevi, İstanbul.
- Korth, Henry F. and Silberschatz, A.**, 1986, Database System Concepts, McGraw-Hill Book Company, University of Texas at Austin, ISBN : 0-07-100529-3.

- Kurbanoglu, S. S.**, Mart 1996, Sanal Gerçeklik : Gerçek mi, Değil mi ?, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 10, Sayı 1, ISSN : 1300-0039.
- Özel, Galip**, Ekim 1992, Hard Disk Yönetim Sistemi, Beta Basım Yayım Dağıtım A.Ş., İstanbul, (1. Baskı), ISBN : 975-486-233-8.
- Özkan, Y. ve Dondurmacı, G. A.**, Kasım 1996, BYTE Türkiye, Oracle Mimarisi, Cilt : 3, Sayı : 11, İhlas Holding A.Ş.
- Tekin, Leyla**, Haziran 1996, Rektörlerin Gözüyle Üniversite Kütüphanelerinin Sorunları, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 10, Sayı 1, ISSN : 1300-0039.
- Uçak, Nazan.Özenç**, Mart 1995, Kütüphaneleri Etkileyen Teknolojik Yenilikler, Türk Kütüphaneciliği Dergisi, Türk Kütüphaneciler Derneği, Cilt 9, Sayı 1, ISSN : 1300-0039.
- Uğur, M.**, April 1991, Use of CSP/AD and CSP/AE on IBM 3090/180S, METU Computer Center Bulletin (CCNEWS), Document No : CC-91.01, Ankara, Turkey.
- Uysal, M.**, Aralık 1994, SQL Veritabanı Sorgulama Dili, Beta Basım Yayım Dağıtım A.Ş., İstanbul, ISBN : 975-486-389-9.
- Voelker, J. A. and Ratica, G. B.**, 1986, The Genesis of a Knowledge-Based Expert System, IBM Systems Journal, Vol. 25, No 2, USA, ISSN : 18-8670.
- Yalçın, F. ve Şahin, F.**, 1993, Fono Yayınları, Açıklamalı Bilgisayar Terimleri Sözlüğü, Ali Bayam (Ed.), Fono Açık Öğretim Kurumu, İstanbul.