

**VERİ KÜPLERİNİN BÜTÜNLEŞİK KULLANIMINA YÖNELİK
YENİ BİR OLAP MİMARİSİ**

Tahsin ÇETİNYOKUŞ

**DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**EYLÜL 2008
ANKARA**

Tahsin ÇETİNYOKUŞ tarafından hazırlanan VERİ KÜPLERİNİN BÜTÜNLEŞİK KULLANIMINA YÖNELİK YENİ BİR OLAP MİMARİSİ adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Hadi GÖKÇEN

Tez Danışmanı, Endüstri Mühendisliği ABD

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Endüstri Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

Prof. Dr. Fevzi KUTAY (Başkan)

Endüstri Mühendisliği ABD, G.Ü.

Prof. Dr. Hadi GÖKÇEN (Danışman)

Endüstri Mühendisliği ABD, G.Ü.

Prof.Dr. Ömer Faruk BAYKOÇ

Endüstri Mühendisliği ABD, G.Ü.

Doç.Dr. İhsan YÜKSEL

Sayısal Yöntemler ABD, KKÜ

Doç.Dr. M. Ali AKÇAYOL

Bilgisayar Bilimleri ABD, G.Ü.

Tarih: 18/09/2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini onamıştır.

Prof. Dr. Nermin ERTAN

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Tahsin ÇETİNYOKUŞ

VERİ KÜPLERİNİN BÜTÜNLEŞİK KULLANIMINA YÖNELİK**YENİ BİR OLAP MİMARİSİ****(Doktora Tezi)****Tahsin ÇETİNYOKUŞ****GAZİ ÜNİVERSİTESİ****FEN BİLİMLERİ ENSTİTÜSÜ****Eylül 2008****ÖZET**

Günümüz veritabanlarında Terabyte(Tb)larca verinin yönetimi olağandır. İnsanlar tarafından kavranamayacak kadar büyük hacimli ve detaylı olan veritabanlarında, özetleme, bütünleştirme ve analiz için; Çevrimiçi Analitik İşleme(OnLine Analytical Processing, OLAP) kullanılmaktadır. OLAP üzerinde yapılan işlemlerin daha hızlı ve kullanılacak modelin daha anlaşılabilir olması için Veri Küplerini kullanmak mümkündür. Küpler, belli bir alanı tariflemesi sebebiyle sorgularda daha hızlı ve iyi netice vermektedir. Pratikte karşılanması imkansız olan basit ihtiyaçlar, mevcut OLAP araçları ile çözülebilmekte fakat büyük veri kümelerinde bu imkansız olmaktadır. Veri küplerinin bu ihtiyacı karşılayacak şekilde kullanımı; çeşitli alanlarda daha avantajlı sistemlerin geliştirilmesine imkan sağlayacaktır. Bu çalışmada, Veri Küplerinin bütünleşerek daha fazla karar desteği sağlamasına yönelik; OLAP mimarisinin bir uzantısı olan yeni bir sistem (Bütünleşik Veri Küpü Sistemi – BVKS) önerilmiştir. Bu sistem sayesinde; OLAP avantajları kullanılarak; Veri Küplerinin kendi aralarında, dışsal verilerle ve çeşitli analitik algoritmalarla ilişkili olduğu bir yapı elde edilmiştir.

Bilim Kodu : 906.1.071
Anahtar Kelimeler : Veri Ambarı(VA), Veri Madenciliği(VM), Çevrimiçi
Analitik İşleme(OLAP), Veri Küpü(VK), Karar Destek
Sistemleri(KDS).
Sayfa Adedi : 120
Tez Yöneticisi : Prof. Dr. Hadi GÖKÇEN

**A NEW OLAP ARCHITECTURE FOR INTEGRATED
USING OF DATA CUBES**

(Ph. D. Thesis)

Tahsin ÇETİNYOKUŞ

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

September 2008

ABSTRACT

In today's databases, management of the data by terabytes is usual. Online Analytical Processing (OLAP) is used for summarizing, integration and analyzing in large volume and detailed databases that can not be envisaged easily by human. Data Cubes (DCs) can be used both to accelerate the processes on OLAP and to make the used model more understandable. Since Cubes define a specific area, they can give faster and effective response in the queries. Simple requirements which are impossible to be met on large volume of data sets in practice can easily be done by available OLAP tools. Usage of Data Cubes for this aim will provide developments of more advantageous systems in various areas. In this study, OLAP based a new architecture is proposed for the purpose of providing decision support at higher level, thereby integrating DCs. For the sake of this architecture, by using OLAP advantages, it is possible to obtain structure in which DCs are related with each other, external data and varied analytic algorithms.

Science Code : 906.1.071

**KeyWords : Data Warehouse, Data Mining, OnLine Analytical Processing,
Data Cube, Decision Support Systems.**

Page Number: 120

Adviser : Prof. Dr. Hadi GÖKÇEN

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Prof.Dr. Hadi GÖKÇEN'e ve tüm çalışma arkadaşlarıma, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	vii
İÇİNDEKİLER	viii
ÇİZELGELERİN LİSTESİ.....	xii
ŞEKİLLERİN LİSTESİ	xiii
SİMGELER VE KISALTMALAR.....	xv
1. GİRİŞ	1
2. VERİ MADENCİLİĞİ.....	4
2.1. Veri Madenciliği Kullanılmasının Gerekçesi	10
2.2. Veri Madenciliğinin Kullanım Alanları.....	11
2.3. Veri Madenciliği Yaklaşımı.....	12
2.4. Veri Madenciliği Tekniklerinin Kullanım Amaçları	13
2.4.1. Sınıflandırma.....	14
2.4.2. Değer biçme	14
2.4.3. Tahmin etme.....	14
2.4.4. Yakınlık veya birlik kuralları	15
2.4.5. Kümeleme	15
2.4.6. Tanımlama ve görselleştirme	15
2.5. Veri Madenciliği Teknikleri	15
2.5.1. Genetik algoritma.....	16
2.5.2. Karar ağaçları	17

Sayfa

2.5.3. Yapay sinir ağıları	18
2.5.4. K-yakın komşuluğu	19
2.5.5. Görüntüleme teknikleri	21
2.5.6. Sorgulama araçları ile veri kümesinin ön analizi	21
2.5.7. OLAP	22
2.6. Veri Madenciliğinin Uygulanabilirliği	23
3. VERİ MADENCİLİĞİ VE OLAP	25
3.1. Veri Ambarı (VA) Kavramı	25
3.1.1. İşletimsel ve bilişimsel sistemler	27
3.1.2. Veri ambarı oluşturma nedenleri	28
3.1.3. Veri çözümleme teknikleri	28
3.1.4. Veri ambarının üretilmesi	29
3.2. Veri Ambarı Mimarisinde Bileşenler	32
3.2.1. İşlevsel veri	32
3.2.2. İşlevsel verideposu (Operational Datastore-ODS)	33
3.2.3. Yük yöneticisi	33
3.2.4. Ambar yöneticisi	33
3.2.5. Sorgu yöneticisi	34
3.2.6. Ayrıntılı veri	34
3.2.7. Az ve çok özetlenmiş veri	34
3.2.8. Arşivlenmiş/Yedeklenmiş veri	34
3.2.9. Meta-Veri	34
3.2.10. Son kullanıcı erişim araçları	35

Sayfa

3.3. Veri Ambarında Veri Akışları	35
3.4. Çok Boyutlu Veritabanı Tasarım Şemaları	36
3.4.1. Yıldız şema.....	37
3.4.2. Kar tanesi şeması.....	37
3.4.3. Takımyıldızı olgusu şema	38
3.5. Veri Tabanı İle Veri Ambarı Karşılaştırması.....	39
3.6. Çevrimiçi Analitik İşleme (OLAP).....	40
3.6.1. OLAP'ın ticari boyutu	41
3.6.2. OLAP'ın özellikleri.....	52
3.6.3. Veri Küpü (VK)	55
3.6.4. Çok boyutlu veri üzerinde OLAP işlemleri	59
3.6.5. OLAP Sunucu Tipleri	62
4. VERİ KÜPLERİNİN BÜTÜNLEŞİK KULLANIMINA YÖNELİK YENİ BİR OLAP MİMARİSİ.....	63
4.1. Literatur Taraması.....	63
4.2. Önerilen Model: Bütünleşik Veri Kübü Sistemi (BVKS)	69
4.3. BVKS'nin Karar Destek Sistemi Yapısı.....	75
5. BÜTÜNLEŞİK VERİ KÜPÜ SİSTEMİ (BVKS) UYGULAMASI.....	77
5.1. Bütünleşik Küp Modulu SATIŞ KÜPÜ Uygulaması	77
5.2. BVKS odaklı OLAP ve VTYS Performans Uygulaması	82
5.3. Uygulama Sonuçlarına Yönelik Deney Tasarımı	92
6. SONUÇLAR	94
KAYNAKLAR	97

Sayfa

EKLER	102
EK-1 BVKS odaklı OLAP ve VTYS Performans Uygulaması Arayuzleri	103
EK-2 BVKS odaklı OLAP ve VTYS Performans Uygulama Program Kodu	108
ÖZGEÇMİŞ	122

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Mevcut VM Teknikleri	22
Çizelge 3.1. Veri tabanı ile veri ambarının karşılaştırılması.....	39
Çizelge 3.2. OLTP ve OLAP karşılaştırması [Han ve Kamber, 2001]	54
Çizelge 5.1. Standart VTYS ile Önerilen BVKS için CPU zamanları karşılaştırma tablosu.....	89
Çizelge 5.2. Standart VTYS ile Önerilen BVKS için CPU zamanları karşılaştırma tablosu	90
Çizelge 5.3. Standart VTYS ile Önerilen BVKS için CPU zamanları karşılaştırma tablosu	91
Çizelge 5.4. Çok değişkenli varyans analizi sonuçları.....	92
Çizelge 5.5. Ürün tipi için Duncan Testi sonuçları.....	93
Çizelge 5.6. Sipariş miktarı için Duncan Testi sonuçları.....	93
Çizelge 5.7. Periyot için Duncan Testi sonuçları.....	93

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2. 1. Veritabanı teknolojisinin gelişimi [Han ve Kamber, 2001]	6
Şekil 2. 2. Bilgi keşfi sürecinde veri madenciliği	8
Şekil 2. 3. Tipik bir veri madenciliği sisteminin yapısı	9
Şekil 2. 4. Veri Madenciliği ve Diğer Disiplinler	13
Şekil 2. 5. Veri Madenciliği kapsamı	16
Şekil 3. 1. Yıldız şema	37
Şekil 3. 2. Kar tanesi şeması	38
Şekil 3. 3. Takımyıldızı olgusu şema	39
Şekil 3. 4. Ürüne göre satış yüzdeleri	43
Şekil 3. 5. Ürün bazında fayda ve hedef skorlar[Pendse, 2006]	44
Şekil 3. 6. Ortalama lisans ücretleri	45
Şekil 3. 7. Değerlendirme sonrası ürün tercih oranları[Pendse, 2006]	46
Şekil 3. 8. Ürüne göre destek kalitesi oranları	47
Şekil 3. 9. Yazılımın uygulamaya geçirilmesi için gerekli iş gücü [Pendse, 2006]...	48
Şekil 3. 10. Uygulamaya geçiş süresi (ay bazında).....	49
Şekil 3. 11. Raporlanan ciddi problemler [Pendse, 2006].....	50
Şekil 3. 12. Performans şikayetleri	51
Şekil 3. 13. Ürünlere göre problem tipi karşılaştırması	52
Şekil 3. 14. Veri küpü örneği [Han ve Kamber, 2001]	55
Şekil 3. 15. Bir üç boyutlu örnek küpün küboidleri	57
Şekil 3. 16. Şekil 3.15'e bağlı olarak Küboitleri gösteren bir çokboyutlu VK	58

Şekil	Sayfa
Şekil 3.17. Çok boyutlu veri üzerinde tipik OLAP işlemleri örneği [Han ve Kamber, 2001].....	59
Şekil 3. 18. Bütünleşik bir OLAM(Çevrimiçi Analitik Madencilik) ve OLAP yapısı [Han ve Kamber, 2001].....	61
Şekil 4. 1. Önerilen modelin getirdiği yeni α küp işlem senaryoları.....	70
Şekil 4. 2. Önerilen modelin getirdiği yeni β küp işlem senaryoları	72
Şekil 4. 3. Önerilen modelin getirdiği yeni σ küp işlem senaryosu	73
Şekil 4. 4. Önerilen modelin getirdiği yeni γ (kompozit yapı) küp işlem senaryoları	73
Şekil 4. 5. BVKS ile elde edilen yeni 3 katmanlı VA mimarisi	74
Şekil 4. 6. a) Temel KDS bileşenleri [Marakas, 1999] b) BVKS'nin KDS bileşenleri	76
Şekil 5. 1. Önerilen mimari ile paket yazılım ve küp girdisiyle, rapor ve yeni küpün oluşumunun uygulamadaki yapısı	78
Şekil 5. 2. Küpü oluşturan boyut tabloları	79
Şekil 5. 3. Satış Küpü Modulu Uygulama Arayuzu.....	80
Şekil 5. 4. Tahmine yönelik regresyon modeli uygulaması	81
Şekil 5. 5. Üretim Küpü boyut hiyerarşisi (a) Makina boyutu (b) Status boyutu	83
Şekil 5. 6. BVKS çizelgeleme uygulaması için kullanılan VT şeması	85
Şekil 5. 7. BVKS çizelgeleme uygulaması arayuzu VTYS çıktısı	86
Şekil 5. 8. BVKS çizelgeleme uygulaması arayuzu OLAP çıktısı	86
Şekil 5. 9. BVKS çizelgeleme uygulaması ile elde edilen GANT çıktısı	88
Şekil 5. 10. Önerilen mimaride α , β durumlarının uygulamadaki yapısı	88

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
α	Sadece veri küplerinin kullanıldığı yapı
β	Veri küpü ve raporların kullanıldığı yapı
σ	Veri küpü ve paket programların kullanıldığı yapı
γ	Kompozit yapı

Kısaltmalar	Açıklama
Tb	Terabyte
OLAP	Çevrimiçi Analitik İşleme
VA	Veri Ambarı
VT	Veri Tabanı
VK	Veri Küpü
VTYS	Veri Tabanı Yönetim Sistemi
BVKS	Bütünleşik Veri Küpü Sistemi
SQL	Yapısal Sorgulama Dili
KDS	Karar Destek Sistemleri
VM	Veri Madenciliği
MD	Çok Boyutlu
OLAM	OLAP Madencilik
OLTP	Çevrimiçi Veri İşleme
MOLAP	Çok Boyutlu OLAP
ROLAP	İlişkisel OLAP
HOLAP	Hibrit OLAP
VTBK	Veritabanlarında Bilgi Keşfi

1. GİRİŞ

Son günlerde her birey ve organizasyon–işletme, aile ya da kuruluş kendisi ya da çevresi ile ilgili büyük miktarda veri ya da bilgiye ulaşabilir. Bu veriler, sistemin dışındaki ilginç değişimlerin ya da eğilimlerin gelişimini tahmin edebilme potansiyeline sahiptir. Fakat, tam anlamıyla başarı çok zaman almaktadır. Bu durum özellikle iş alanında böyledir ve başlıca iki adet büyük problem bulunmaktadır. Bilgi farklı dosyalama sistemlerine dağılmıştır. Bu sistemler bir diğeriyle bağlı değildir ve verimsiz bir organizasyon ile veriler üretilmektedir. Bilgilerin detaylandırılmasında istatistiksel araçlar ve onların gücü ile ilgili yeterli bilgiye sahip olunmayışı diğeri bir problemdir. Bu durum verinin sentezi ve verinin verimli üretiminde karşımıza çıkmaktadır [Giudici, 2003].

İfade edilen bu problemler ile başa çıkabilmemizde iki gelişme fayda sağlayacaktır: Birincisi; organizasyonların verileri toplaması, daha kolay erişimi ve transferi için yazılım ve donanımın daha az maliyetle, daha fazla güç üretimini sağlamasıdır. İkincisi; son günlerde büyük veri depolarının analizindeki, özellikle bilgisayar ve istatistik alanındaki metodolojik araştırmalar sonucu meydana gelen gelişmeler ve esnek, ölçeklenebilir süreçlerin kullanılabilirliğidir. Bu iki gelişme, veri madenciliği (VM)'nin karar desteği için, iş alanlarına hızlı şekilde ve önemli bir zeki araç olarak yayılması anlamına gelmektedir.

Veri oluşturma ve toplama teknikleri hızla ilerlemektedir. Şu an ki aşamada, veri eksikliği artık sorun (verilerden yararlı bilgi oluşturma yetersizliği) olmaktan çıkmıştır [Sang ve Keng, 2001].

Veriyi yararlı/kullanılabilir bilgiye, akıllıca ve kendiliğinden dönüştürmek için yeni teknolojilerin ve araçların geliştirilmesine ihtiyaç, veri ve veritabanı (VT) alanlarındaki patlama yapan gelişmenin sonucunda doğmuştur. Gelişmelerin ve ihtiyaçların neticesinde büyük veri kümelerinde desen çıkarımında kullanılan VM, zamanla artan bir önemle bir araştırma alanı/konusu haline gelmiştir [Weiss ve

Indurkha, 1998; Waterhouse, 1997; Fayyad ve ark., 1996; Piatetsky-Shapiro ve Frawley, 1991].

Modern VT için Terabyte ^{*} (Tb)larca verilerin yönetimi olağandır. Winter şirketinin 2001 yılı araştırmalarına göre, örneğin; bir telefon firması olan SBC'nin Karar Destek VT 10.5 Tb'a ulaşmıştır. Dahası, incelenen şirketlerin %17 si 3 yıl içinde kendi VT on katına çıkacağını düşünmektedirler. Uzmanlar detaylı seviyeleri olan bu kadar büyük hacimli veri/bilgi nin, insanlar tarafından kavranamayacağını belirtmişlerdir. Onun yerine, onlara sistem üzerinde verilerin bir kısmı için özetleme ve belirli görevlerin yerine getirilmesinde itimat edilebileceği söylenmiştir. Sonuç olarak, verimli özetleme ve büyük hacimli verilerin bütünleştirilmesinde Çevrimiçi Analitik İşleme (OnLine Analytical Processing-OLAP) ve Karar Destek Sistemleri (KDS) nin önemli rol oynayacağı söylenebilir. Çoğu bütünleşmiş bilginin bakımı, kavramsal modeli daha kolay anlaşılan veri küplerinin içinde yapılmaktadır. Bunun sebebi, sorguların veri kübü içindeki belli bir bölgeyi tarif edebilmesidir. Önceden hesaplanmış ya da tekrarlı bütünleşik verilerdeki gibi, sorgu işleme hızını artıran grublama, ara tablolar ve alt hesaplamalar KÜP operatörünün önerildiği gibi kolayca modele dahil edilebilirler [Gray ve ark., 1997].

OLAP kullanımının asıl amacı, sorgularda olduğu gibi VT'nda veri analizini desteklemektir. Güncelleştirilme maliyetleri böylece önemsiz hale gelecektir. Sonuçta, gecelik işlemlerde olduğu gibi ambarlara düşük iş yüklemeleri yapıldığında sistemler küme halinde güncellemeye yönelecektir. Çoğu uygulamada özellikle karar destek ve stok yönetiminde son bilgilerin olabildiğince erken alınması karar avantajı sağlar. Yine de OLAP doğal olarak What-If senaryolarını kapsayan interaktif bilginin çıkarımı sürecini gerçekleştirir. What-If analizleri geçmiş verilerin anlık analizi ve sonraki faaliyetler için birleştirilmiş gerçek zaman verilerine ihtiyaç duyar. Analizi daha verimli ve hızlı bir şekilde yapmak için veri küplerinin kullanımı uygun olacaktır.

^{*} 1 Tb=1024 Gigabyte

Bu tezde, çok seçenekli esnek raporlama ve çok hızlı sorgu performansı elde etmek amacıyla; OLAP mimarisinin uzantısı olan yeni bir sistem önerilmiştir. Mevcut OLAP işlemleri sınırlı yetenekler sunmaktadır. Bu sistem sayesinde; OLAP avantajları kullanılarak; her alanda problem çözümüne imkan sağlayan esnek yapıları bir mimari sunan, Veri Küplerinin kendi aralarında, dışsal verilerle ve çeşitli analitik algoritmalarla ilişkili olduğu bir yapı elde edilmiştir.

Bilinebildiği kadarıyla literatürde benzeri bir yaklaşım bulunmamaktadır. Veri Küpü kullanımına yeni bir yaklaşım getiren çalışmanın literatürde katkısı bulunmaktadır.

Çalışmanın birinci bölümünde tezin konusu, amacı, temel problem ve problemin hangi yöntemle çözüleceği verilmiştir. İkinci bölümünde Veri Madenciliği başlığı altında, kullanım gerekçesi, teknikleri ve Veri Madenciliğinin uygulanabilirliği yer almıştır. Üçüncü bölümde Veri Madenciliği ve OLAP ilişkisinden bahsedilerek, Veri Ambarı kavramı, bileşenler ve OLAP açıklanmıştır. Önerilen Bütünleşik Veri Küpü Sistemi Mimarisi dördüncü bölümde yer almıştır. Önerilen mimarinin test edilmesine yönelik uygulamaların bulunduğu beşinci bölümden sonra altıncı bölümde sonuçlar yer almıştır.

2. VERİ MADENCİLİĞİ

Veri madenciliği terimini anlamak için kelime anlamına bakmak faydalı olacaktır: madencilik, seçip çıkarmak anlamındadır. Fiili olarak dünyadan kıymetli kaynaklar çıkarılması işlemidir. Veri kelimesi ile birleştirildiğinde; veri yığınları içinde önceden bilinemeyen ek bilgilerin bulunabilmesi için derin araştırma akla gelmektedir. Bilimsel araştırmaların bakış açısıyla veri madenciliği, bilgisayar hesaplamaları, pazarlama ve istatistik gibi diğer disiplinlere nazaran yeni gelişen bir disiplindir. Veri madenciliğinde kullanılan metodolojilerden bir çoğu iki kısma ayrılmıştır. Birisi; makina öğrenimi, diğeri ise özellikle çok değişkenli ve sayısal istatistiği kapsayan istatistik alanıdır [Giudici, 2003].

Makina öğrenme, bilgisayar bilimleri ve yapay zeka ile ilgilidir aynı zamanda genel doğruları belirlenen verilerin arasındaki ilişki ve kuralların bulunması ile ilgilidir. Makina öğreniminde; veri üretici süreçlerle yeniden veri üretiminde analiste gözlenen değerlerden, yeni ve gözlenmemiş durumları elde etmeye imkan sunulması amaçlanır. Rosenblatt, 1962’de “perception” adıyla anılan ilk makina öğrenme modelini geliştirmiştir. Bunun arkasından 1980 lerin ikinci yarısına doğru, yapay sinir ağları geliştirilmiştir. Bu sıralarda, bazı araştırmacılar esas olarak sınıflandırma problemleri üzerinde çalışacak karar ağaçları teorisini mükemmelleştirmişlerdir. 1980 lerin ikinci yarısından sonra istatistiksel analiz temelinde bilgisayar kullanılarak uygulanan yöntemlere verilen önem artmıştır. Bununla birlikte çok değişkenli uygulamaların analizinde de paralel gelişmeler olmuştur. 1990 larda istatistikçiler makine öğrenme yöntemlerine ilgi göstermeye başlamışlar ve metodolojide önemli gelişmeler sağlanmıştır [Giudici, 2003].

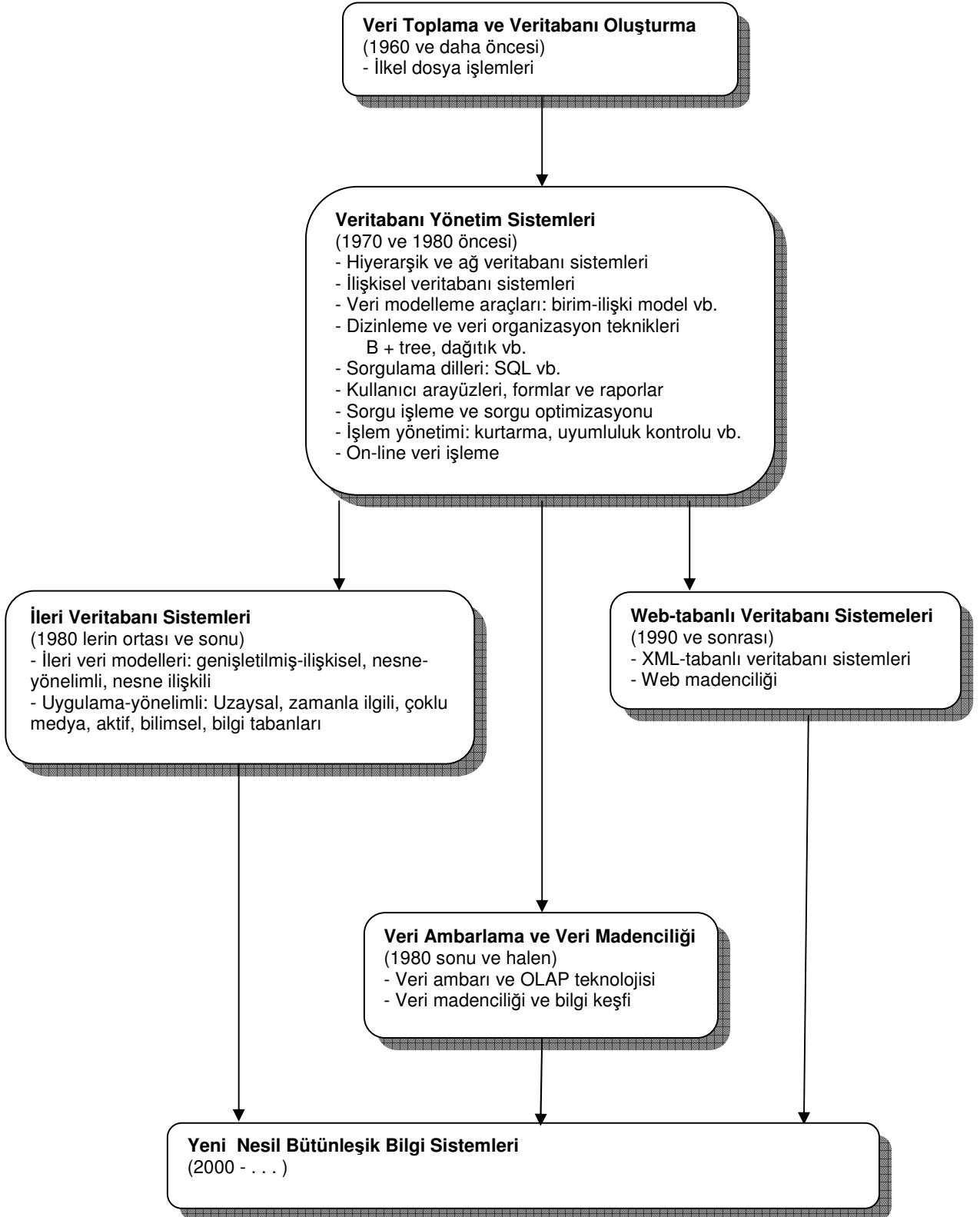
1980 lerin sonuna doğru makine öğrenme yöntemleri bilgisayarlı hesaplama (iteratif işlemler) ve yapay zeka alanları dışında da kullanılmaya başlanmıştır. Özellikle ayrıntılı ve özel pazarlama kampanyalarında kullanılan mevcut veritabanlarının da içinde olduğu veritabanı pazarlama uygulamalarında kullanılmıştır. Veritabanlarında bilgi keşfi, gözlenen veriler arasındaki ilişkileri ve kuralları bulma amacıyla olan tüm bu yöntemleri tarif etmektedir. Yavaş yavaş veritabanı bilgi keşfi,

veritabanından bilgi çıkarımı süreçlerinin tarifi ve anlaşılması ile genişlemiştir. Veri madenciliği terimi, verilere öğrenme algoritmalarının uygulanması süreçlerinin bileşenlerinden tariflenmiştir. Şekil 2.1’de veritabanı teknolojilerinin gelişimi görülmektedir [Han ve Kamber, 2001].

Terminolojide, VM ilk defa Usama Fayyad tarafından I. Uluslararası Bilgi Keşfi ve Veri Madenciliği konferansında 1995 te Montreal’de ortaya konmuştur ve hala bu başlık altında tek ana konferans olduğu düşünülmektedir. Önemli ilişkileri ya da düzenleri açıkça görülemeyen eldeki büyük veri kümelerinden, önceden bilinmeyen bilgilerin ortaya çıkarılması amacı ile bütünleşik analitik tekniklerin birçok aşamaya bölünmesinde de referans olarak kullanılmıştır. Kelime olarak veri madenciliği yavaş yavaş kendi tarifini yapmıştır ve bilgi çıkarımında kullanılan tüm süreçlerle aynı anlama gelmeye başlamıştır. Geniş bir tanımlama yapılacak olursa;

“Veri madenciliği, veritabanı sahibine anlaşılır ve faydalı sonuçlar vermek amacıyla, büyük miktardaki verilerin daha önceden bilinmeyen ilişki ve kuralların keşfedilebilmesi için modelleme, çıkarım ve seçim sürecidir.” [Giudici, 2003]

Bir veri madenciliği yöntemini uygulamak, işin gereksinimlerini probleme uyarlayarak bütünleşik bir yöntemin kullanılması demektir. Bunun için, problemin analizi, veritabanı gereksinimlerinin sağlanması ve stratejik kararın alınabileceği son hedef için başarılı, önemli sonuçlar veren, bilgisayarda uygulanabilecek istatistiksel tekniklerin kullanılması gerekmektedir. Stratejik karar kendine özgü yeni ölçülere gerek duyacaktır ve sonuç olarak veri madenciliğinin eyleme geçirdiği bilgilerin faydalı çevrimi olarak adlandırılan yeni iş gereksinimlerini de beraberinde getirecektir [Berry ve Linoff, 1997].



Şekil 2. 1. Veritabanı teknolojisinin gelişimi [Han ve Kamber, 2001]

Veri madenciliği sadece istatistiksel tekniklerin ya da hesaplama algoritmalarının kullanımı değildir, aynı zamanda firma kararlarını destekleyen bilgi teknolojilerini sağlayan iş zekasının bir sonucudur.

Konunun önde gelen uzmanlarından Piatetsky-Shapiro da veri madenciliğini, verilerden daha önceden bilinmeyen, gizli, muhtemelen faydalı enformasyonun monoton olmayan bir süreçte çıkartılması işlemi olarak tanımlamaktadır. Bu süreç kümeleme (*clustering*), veri özetleme (*data summarization*), sınıflama kurallarının (*classification rules*) öğrenilmesi, bağımlı ağların (*dependency networks*) bulunması, değişikliklerin analizi (*analysing changes*) ve anomali tespiti (*detecting anomaly*) gibi farklı bir çok teknik yaklaşımı kapsamaktadır [Dilly, 2005].

Gartner Group tarafından yapılan bir diğer tanımda ise veri madenciliği, istatistik ve matematik tekniklerle birlikte örüntü tanıma (*pattern recognition*) teknolojilerini kullanarak, depolama ortamlarında saklanmış bulunan veri yığınlarının elenmesi ile anlamlı yeni korelasyon, örüntü ve eğilimlerin keşfedilmesi sürecidir [SPSS, 2005].

Veri madenciliği tanımları oldukça fazladır. Friedman (1997) veri madenciliği literatüründeki tanımları bir araya getirmiştir [Kovalerchuk, 2000]:

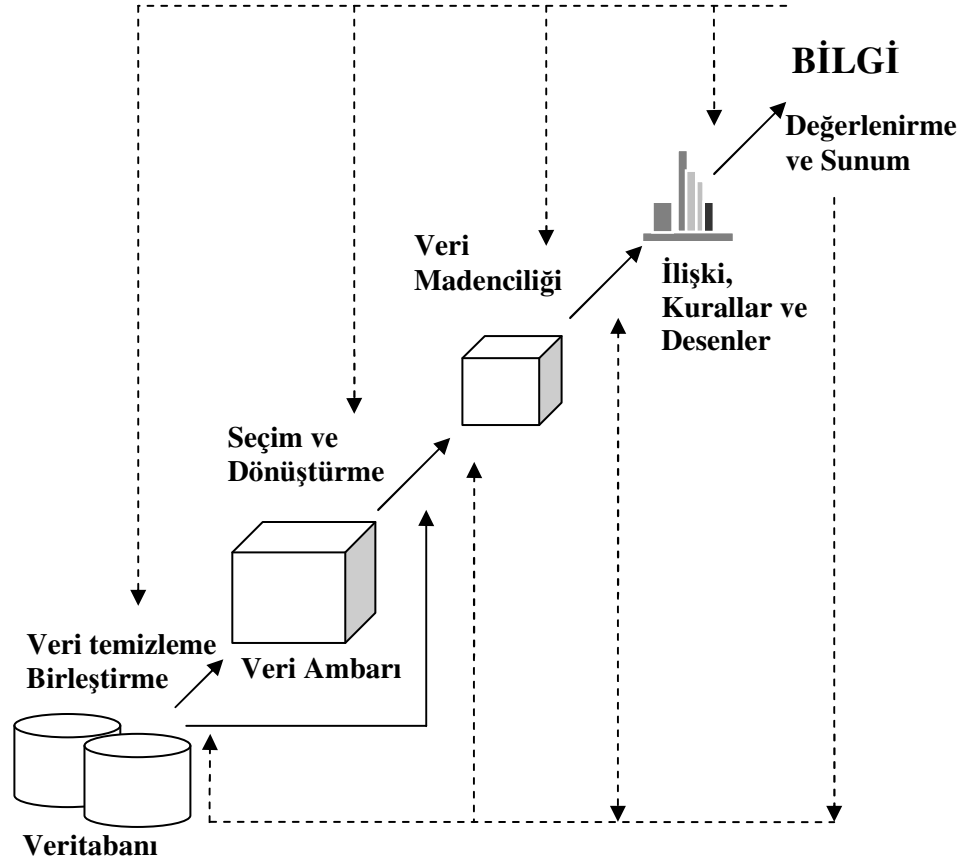
Fayyad’a göre veri madenciliği; verilerden, geçerli, alıılmamış, potansiyel olarak kullanışlı ve sonuçta anlaşılabilir desenlerin, ilişkilerin tanımlanması için önemli bir süreçtir.

Zekulin’e göre veri madenciliği önceden bilinmeyen, anlaşılabilir ve dikkate değer bilgilerin büyük veri tabanlarından çıkarım sürecidir ve çok önemli işletme kararlarının alımında kullanılır.

Ferruza’ya göre veri madenciliği, veriler içinden önceden bilinmeyen ilişkileri ve desenleri ayırt edebilmek için bilgi keşfi sürecinde kullanılan bir yöntemler kümesidir.

Parseye'ye göre veri madenciliği, büyük veri tabanlarında bilinmeyen ve beklenmeyen desenleri araştırırken bilgi veren bir karar destek sürecidir.

Bir çok kişi veri madenciliği ile diğer popüler terim olan veritabanlarında bilgi keşfini eşanlamlı kabul ederler. Alternatif olarak diğerleri; veri madenciliğini, veritabanında bilgi keşfi sürecinde adımlardan biri olarak kabul ederler. Şekil 2.2 'de bilgi keşfi bir sürecinde tariflenen ve takip eden adımlarda sıralı dizinin birbileşeni olan veri madenciliği görülmektedir.

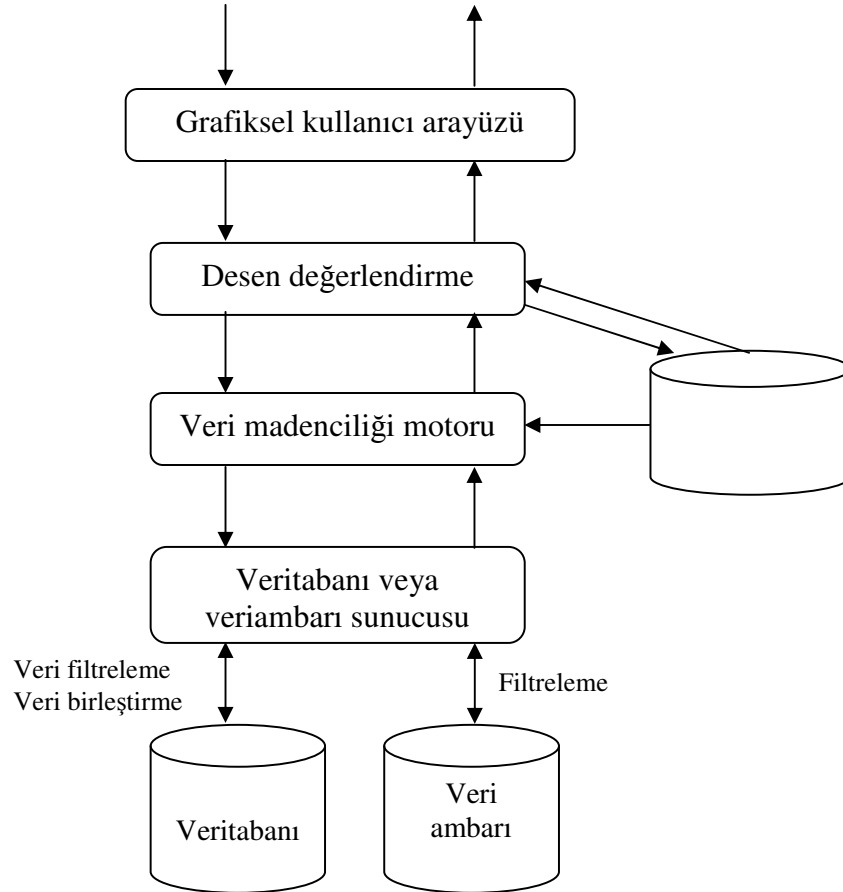


Şekil 2. 2. Bilgi keşfi sürecinde veri madenciliği [Han ve Kamber, 2001]

1. Veri temizleme (parazit ve tutarsız olanların ayıklanması)
2. Veri birleştirme (çoklu veri kaynağı olduğunda birleştirilmesi)
3. Veri seçimi (veritabanından, analiz ile ilişkili verilerin seçilmesi)

4. Veri dönüştürümü (bütünleşik operasyonlar ya da özetleme faaliyetleri tarafından madencilik için dönüştürme ve birleştirme işlemi)
5. Veri madenciliği (veri desenlerin çıkarımı için sıralı uygulanan zeki metodların bir süreci)
6. Desen değerlendirme (bilgi tabanlı bazı ölçütleri temsil eden ilginç desenlerin tam olarak tanımlanması)
7. Bilgi sunumu (kullanıcılara çıkarılan sonuçların görsel ve bilgi sunum teknikleri kullanılarak aktarılması)

Şekil 2.3 'de ise tipik bir veri madenciliği sisteminin yapısı görülmektedir [Han ve Kamber,2001].



Şekil 2. 3. Tipik bir veri madenciliği sisteminin yapısı

2.1. Veri Madenciliği Kullanılmasının Gerekçesi

Günümüz modern insanının her alışverişinde, her bankacılık işleminde, her telefon edişinde kaydedilen, uzaktan algılayıcılardan, uydulardan toplanan, devlet ve işletme yönetiminde yapılan işlemler sonucunda saklanan veriler, her an inanılmaz boyutlarda artmaktadır. Sadece uydu ve diğer uzay araçlarından elde edilen görüntülerin saatte 50 GB düzeyinde olması, bu artışın boyutlarını daha açık bir şekilde göstermektedir.

Bu verilerin elde edilmesini kolaylaştıran otomatik veri toplama araçları ve veri tabanı teknolojilerindeki gelişme, veritabanlarında, veri ambarlarındaki ve diğer bilgi depolarındaki bilgilerin hızlı bir şekilde kendisini katlamasına olanak sağlamış; geleceği veri karanlığına dönüştürebilecek çok miktarda ama sonuç üretilmeyen bilgi depolanmasına neden olmuştur.

1995 yılında birincisi düzenlenen Veritabanlarında Bilgi Keşfi((VTBK)Knowledge Discovery in Databases) konferansında, enformasyon teknolojilerinin oluşturduğu veri dağları aşağıdaki cümleler ile vurgulanmaktadır [Berry ve Linoff, 1997].

“Dünyadaki enformasyon miktarının her 20 ayda bir ikiye katlandığı tahmin edilmektedir. Bu ham veri seli ile ne yapmamız gerekmektedir? İnsan gözleri bunun ancak çok küçük bir kısmını görebilecektir. Bilgisayarlar bilgelik pınarı olmayı vaat etmekte, ancak veri sellerine neden olmaktadır.”

Güney Amerikalı yazar Jorge Louis Borges’in hikayelerinden biri olan Babil kütüphanesini düşünelim. Bu kütüphane sonsuz raflı odalardan oluşur ve bu odada içerikleri bilinmeyen sonsuz sayıda kitap vardır. Bilim adamları bu kütüphaneden insanların yararlanabilmesi için yöntemler geliştirmek üzere hipotezler kurmuşlardır.

Bu hipotezler:

- Kütüphanenin bir merkez kataloğunun bulunması gerekir.

- Kütüphanedeki her bir kitabın kütüphanenin herhangi bir yerinde olduğu düşünülebilir.
- Kütüphane kendisini sonsuz bir şekilde tekrar eden sınırsız bir sayıya sahiptir.

Ama bu hipotezlerin hiçbiri doğrulanamaz. Kütüphane sonsuz sayıda veriye sahiptir fakat bu verilerden bilgi elde etmek çok zordur [Adriaans ve Zantinge, 1996].

2.2. Veri Madenciliğinin Kullanım Alanları

Veri madenciliği, astronomi, biyoloji, finans, pazarlama, sigorta, tıp ve bir çok başka dalda uygulanmaktadır. Son 20 yıldır Amerika Birleşik Devletleri'nde çeşitli veri madenciliği algoritmalarının gizli dinlemeden, vergi kaçakçılıklarının ortaya çıkartılmasına kadar çeşitli uygulamalarda kullanıldığı bilinmektedir [Dilly, 2005].

Bununla birlikte günümüzde veri madenciliği teknikleri özellikle işletmelerde çeşitli alanlarda başarı ile kullanılmaktadır. Bu uygulamaların başlıcaları ilgili alanlara göre şöyle özetlenmiştir [SAS, 2005].

Pazarlama;

- Müşterilerin satın alma örüntülerinin belirlenmesi,
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması,
- Posta kampanyalarında cevap verme oranının artırılması,
- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması,
- Pazar sepeti analizi (*Market Basket Analysis*)
- Müşteri ilişkileri yönetimi (*Customer Relationship Management*)
- Müşteri değer analizi (*Customer Value Analysis*)
- Satış tahmini (*Sales Forecasting*)

Bankacılık;

- Farklı finansal göstergeler arasında gizli korelasyonların bulunması,
- Kredi kartı dolandırıcılıklarının tespiti,
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi,
- Kredi taleplerinin değerlendirilmesi.

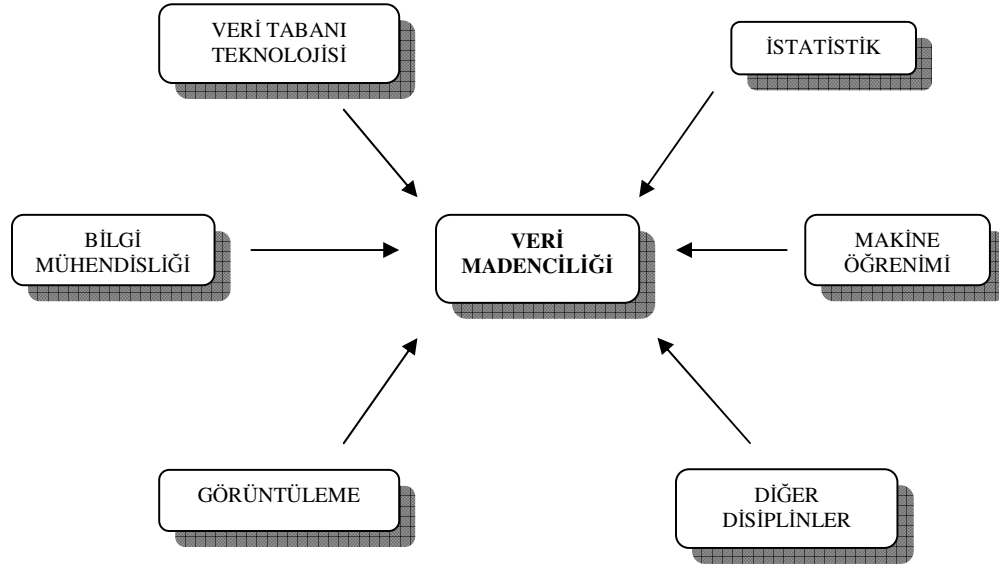
Sigortacılık;

- Yeni poliçe talep edecek müşterilerin tahmin edilmesi,
- Sigorta dolandırıcılıklarının tespiti,
- Riskli müşteri örüntülerinin belirlenmesi.

2.3. Veri Madenciliği Yaklaşımı

Veri madenciliği, veri tabanı sistemleri, verilerin depolanması, istatistik, makine öğrenimi, bilgi düzeltme ve yüksek performanslı hesaplamalar gibi alanlardan çıkan genç bir disiplinler arası alandır (Bkz. Şekil 2.4). Diğer yardımcı alanlar, sinir ağlarını, kimlik belirleme, uzaysal veri çözümlemesini, olasılık grafik teorisini ve tümevarımsal mantık programlamasını içerir. Veri madenciliği birden fazla disiplinden elde edilen yaklaşımların bütünleştirilmesine ihtiyaç duyar [Hand, 1999].

Verileri çözümleme metotları istatistikte yıllarca yapılan çalışmalarla geliştirilmiştir. Makine öğrenimi, sınıflama ve tümevarım problemlerinin çözümünde etkilidir. Yapay sinir ağları ise veri sınıflama, tahmin ve kümeleme çözümlemesi konularında etkinliğini göstermiştir. Fakat veri madenciliği, veri tabanında gittikçe artan büyük miktarda veriler depolandığında, bu yöntemleri, hesaplanabilirlik ve etkinliği açıklamada kullanır. VT araştırmalarında etkin veri yapıları, indeksleme ve veri girişi teknikleri veri madenciliğinin performansını yükseltmek amacıyla geliştirilmektedir [Hand, 1999; Glymour ve ark., 1996].



Şekil 2. 4. Veri madenciliği ve diğer disiplinler

2.4. Veri Madenciliği Tekniklerinin Kullanım Amaçları

Veri madenciliğinde birçok farklı yöntem ve teknik kullanılmaktadır [Hand, 1999].

Bu tekniklerin kullanım amaçları 6 ana başlıkta ifade edilmektedir. Bunlar:

- i) Sınıflandırma
- ii) Değer biçme
- iii) Tahmin etme
- iv) Yakınlık veya birlik kuralları
- v) Kümeleme
- vi) Tanımlama ve görselleştirme dir

Bunlara ek olarak kavramların tanımlanması, eksik değerleri tahmin gibi amaçlarla da kullanılmaktadır.

Bu ana başlıkların ilk 3 tanesi, yönlendirilmiş veri madenciliğine örnektir. Burada amaç, bir model kurmak için, incelenmek istenen bir değişkeni geri kalan verilerle tanımlamaktır. Diğer 3 ana başlık ise yönlendirilmemiş veri madenciliğine örnektir.

Burada hedef olarak belirlenen bir deęişken bulunmamakta, bütün deęişkenler arasında bir ilişki kurmak amaçlanmaktadır.

2.4.1. Sınıflandırma

Sınıflandırma, yeni bir nesnenin niteliklerini inceleme ve bu nesneyi önceden tanımlanmış bir sınıfa atamaktır. Burada önemli olan, her bir sınıfın özelliklerinin önceden net bir şekilde belirlenmiş olmasıdır. Sınıflandırmaya örnek olarak kredi kartı başvurularını düşük, orta ve yüksek risk grubu olarak ayırmak veya segmentasyon yapmak gösterilebilir.

2.4.2. Deęer biçme

Sınıflandırmanın aksine deęer biçme yöntemi, kredi kartı bakiyesi ve gelir düzeyi gibi sürekli deęişkenlik gösteren sonuçlar ile ilgilidir. Aslında deęer biçme yöntemi sınıflandırma yapmada bir araç olarak kullanılır. Örneğin müşterilerine bir teklif götürmek isteyen firmalar, daha önceki tekliflerine müşteriler tarafından verilen yanıtları puanlayarak, 0 ile 1 arasında müşterilerini derecelendirir ve olumlu yanıt vermesi en olası müşterilere teklifini sunar.

Bir dięer örnek, müşteri olmayı bırakma olasılığı yüksek olan müşterilere deęer biçmektir (churn modelling). Daha önce firmayı terk etmiş müşterilerin ortak özelliklerini belirleyerek, bu özelliklere sahip olan mevcut müşterilere yeni teklifler ve özel kampanyalar sunarak bu müşterilerin rakip firmalara yönelmeleri önlenabilir.

2.4.3. Tahmin etme

Tahmin etme yöntemini, sınıflama ve deęer biçme yöntemlerinden ayrı tutmamak gerekmektedir. Çünkü bu yöntemler elde bulunan verilerle yapılan tahminlere dayanır. Bu tahminlerin gerçekleşip gerçekleşmedięi ise ancak bekleyerek görülebilir. Örneğin geçmiş verilere dayanılarak 6 ay içinde firmayı terk edeceğini düşündüğümüz bir müşteri bunun aksi yönde hareket edebilir. Bu veri de gelecekte

yapılacak olan tahminler için kullanılacaktır.

2.4.4. Yakınlık veya birlik kuralları

Burada amaç, beraber giden şeyleri belirlemektir. Klasik bir örnek olarak, bir süper markette beraber satılan ürünleri düşünebiliriz. Perakendeciler bu bilgiyi raf düzenlemeleri, katalog tasarlaması ve çapraz-satış fırsatları yaratmak için kullanmaktadır.

2.4.5. Kümeleme

Kümeleme, değişik özellikler gösteren insanları, kendi içinde benzerlik gösteren alt gruplara ve kümelere ayırma işlemidir. Sınıflamadan farkı ise, kümelemede özellikleri önceden belirlenmiş sınıfların varolmayışıdır. Ortaya çıkan kümelere, eğer varsa, bir anlam yüklemek, analizi yapan kişiye bağlıdır. Örneğin birbirinden farklı müzik tarzlarını beğenen kişiler kümesi, aynı zamanda farklı alt kültür gruplarını da gösteriyor olabilir.

2.4.6. Tanımlama ve görselleştirme

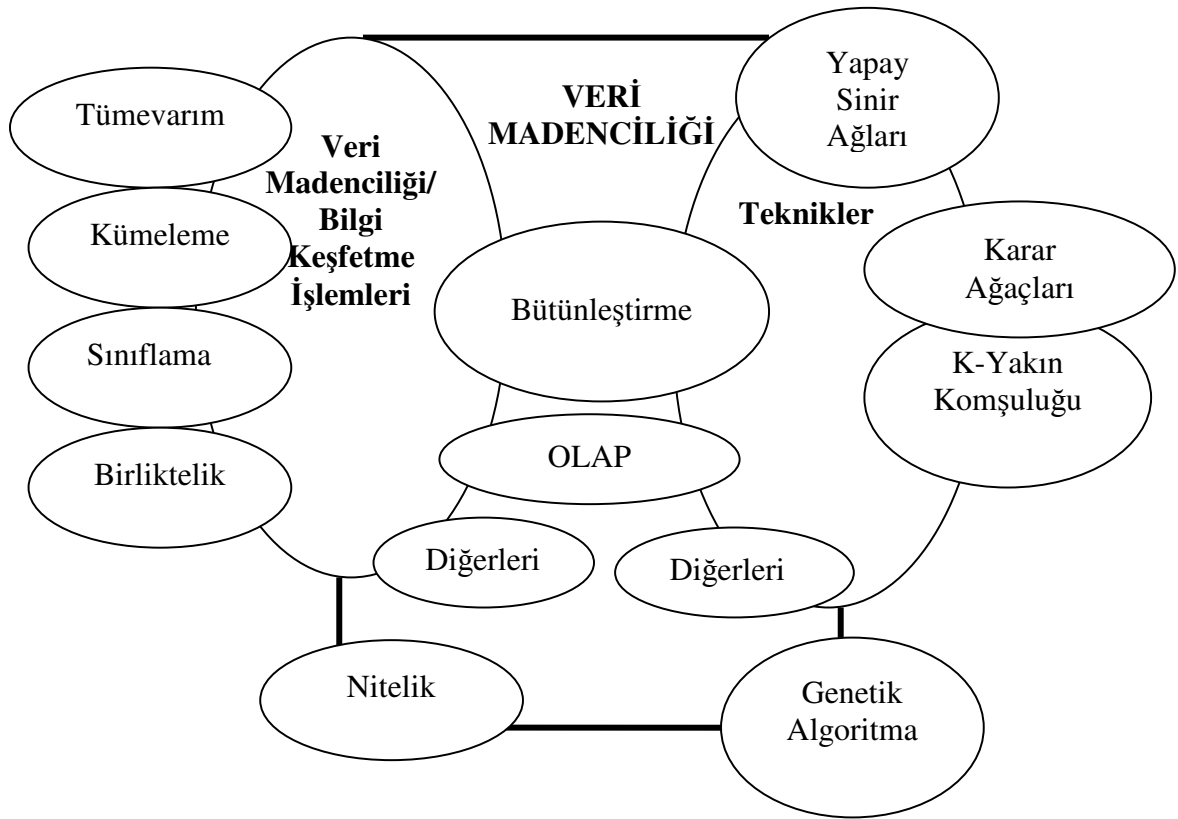
Veri madenciliğinin bazen amacı, sadece karmaşık bir veritabanında neler olup bittiğini anlaşılır kılmaktır. İyi bir tanımlama, en azından, nereden başlamak gerektiği konusunda yol gösterici olabilir. Bulunan tanımlamaları göze hitap edecek bir şekilde anlamlandırabilmek de binlerce ilişki bulmaktan çok daha doğru bir resim ortaya koyabilir.

2.5. Veri Madenciliği Teknikleri

Veri madenciliği, veri kümelerinden gizli kalmış bilgilerin çıkarıldığı tek başına bir teknik değildir. Çok farklı tekniklerin birlikte kullanıldığı bir işlemdir (Bkz. Şekil 2.5)

Veri madenciliğinde kullanılan bazı önemli teknikler şunlardır.

- Genetik Algoritma
- Karar Ağaçları
- Yapay Sinir Ağları
- K-Yakın Komşuluğu
- Görüntüleme Teknikleri
- Sorgulama Araçları ile Veri Kümesinin Ön Analizi
- OLAP



Şekil 2. 5. Veri madenciliği kapsamı

2.5.1. Genetik algoritma

Evrimsel hesaplama ilk olarak 1960'larda I.Rechenberg tarafından "Evrimsel Stratejileri (Evolutions strategie)" isimli eserinde tanıtılmıştır. Onun fikri daha sonra

başka araştırmacıların da ilgisini çekmiş ve geliştirilmiştir. John Holland evrim süreci kullanılarak, bilgisayara anlayamadığı çözüm yöntemlerinin öğretilbileceğini düşünmüş ve Genetik Algoritma (GA) bu düşüncenin bir sonucu olarak bulunmuştur.

2.5.2. Karar ağaçları

Gerçek dünyanın sosyal ve ekonomik olaylarını daha güvenilir bir şekilde gösterebilmek için standart istatistiksel tekniklerin dışında yeni analiz tekniklerinin geliştirilmesi ile ilgilenen Morgan ve Sonquist tarafından 1970’li yılların başlarında Michigan Üniversitesinde kullanıma alınan *Automatic Interaction Detector – AID* karar ağacı temelli ilk algoritma ve yazılımdır. AID tekniği en kuvvetli ve en iyi tahmini gerçekleştirebilmek için bağımlı ve bağımsız değişkenler arasındaki mümkün bütün ilişkilerin incelenmesine dayanmaktadır. Karar ağacı tekniğinin sağladığı kuruluş ve yorumlama kolaylıkları, AID yazılımının başlangıçta istatistikçi ve veri analistleri tarafından büyük coşku ile karşılanmasına neden olmuştur.

Tahmin edici ve tanımlayıcı özelliklere sahip olan karar ağaçları, veri madenciliğinde

- Kuruluşlarının ucuz olması,
- Yorumlanmalarının kolay olması,
- Veri tabanı sistemleri ile kolayca entegre edilebilmeleri,
- Güvenilirliklerinin daha iyi olması

nedenleri ile sınıflama modelleri içerisinde en yaygın kullanıma sahiptir.

Karar ağacı temelli analizlerin yaygın olarak kullanıldığı sahalar [SPSS, 1998],

- Belirli bir sınıfın muhtemel üyesi olacak elemanların belirlenmesi (Segmentation),
- Çeşitli vakaların yüksek, orta, düşük risk grupları gibi çeşitli kategorilere ayrılması (Stratification),

- Gelecekteki olayların tahmin edilebilmesi için kurallar oluşturulması,
- Parametrik modellerin kurulmasında kullanılmak üzere çok miktardaki değişken ve veri kümesinden faydalı olacakların seçilmesi,
- Sadece belirli alt gruplara özgü olan ilişkilerin tanımlanması,
- Kategorilerin birleştirilmesi ve sürekli değişkenlerin kesikliye dönüştürülmesidir.

Karar ağacı temelli tipik uygulamalar ise [SPSS, 1998],

- Hangi demografik grupların mektupla yapılan pazarlama uygulamalarında yüksek cevaplama oranına sahip olduğunun belirlenmesi (Direct Mail),
- Bireylerin kredi geçmişlerini kullanarak kredi kararlarının verilmesi (Credit Scoring),
- Geçmişte işletmeye en faydalı olan bireylerin özelliklerini kullanarak işe alma süreçlerinin belirlenmesi,
- Tıbbi gözlem verilerinden yararlanarak en etkin kararların verilmesi,
- Hangi değişkenlerin satışları etkilediğinin belirlenmesi,
- Üretim verilerini inceleyerek ürün hatalarına yol açan değişkenlerin belirlenmesidir.

2.5.3. Yapay sinir ağları

Yapay sinir ağları öğrenme yeteneğine sahip, gelişmiş matematiksel yapıların hesaplanmasını içeren bir yaklaşımdır. Bu metot sinir sisteminin öğrenmesini model alan akademik araştırmaların bir sonucu olarak ortaya çıkmıştır. Sinir ağları karmaşık ve anlaşılması çok güç olan yapılardan anlam türetme becerisine sahip, dikkate değer yeteneklere sahiptir. Sinir ağları, çok karmaşık yapıların eğilimlerinin hem manuel tekniklerle hem de bilgisayar teknikleriyle analizine imkan sağlamıştır. İyi kurulmuş bir sinir ağı, analiz edilmesi amacıyla bilgilerin verildiği bir uzman olarak düşünülebilir. Bu uzman yeni durumların ilginç yanlarının ve “eğer ne olursa/değişirse/yapılmışsa?” gibi sorularının cevaplanmasını sağlar.

Yapay sinir ağlarının gerçek iş hayatıyla ilgili problemler için çok geniş uygulanabilirliği bulunmaktadır. Uzun zamandır pek çok sanayi kuruluşunda başarıyla uygulanmaktadır. Yapay sinir ağlarının verilerin eğilimlerinin belirlenmesindeki başarısı nedeniyle, aşağıdaki tahmin ve öngörüler için en kullanışlı araçtır.

- Satış tahminleri
- Endüstriyel işlem kontrolleri
- Müşteri araştırmaları
- Verilerin geçerliliklerinin doğrulanması
- Risk yönetimi
- Hedef pazarlama

Yapay sinir ağları işlem elemanlarının (ya da düğümlerin) beyin nöronlarına benzetilerek çözülmesine yönelik düzenlemeler olarak değerlendirilir. Bu elemanlar bir ağ içinde birbirlerine bağlıdır. Ağla oluşturulan modellerin belirlendiği verilere bir defa girilince ağ, insanların yaptığı gibi önceki deneyimlerinden kendi öğretilerini oluşturmaya başlar. Bu özellik yapay sinir ağlarını geleneksel hesaplama programlarından ayıran, onu sabit ardışık sırada basit bir izleme yönergesi yapan özelliğidir.

2.5.4. K-yakın komşuluğu

Kayıtlar, bir veri uzayındaki noktalar olarak düşünülürse, birbirine yakın olan kayıtlar, birbirinin civarında (yakın komşusu) olur.

Müşteri kayıtlarının bulunduğu bir veri tabanından, müşterilerin davranışları tahmin edilmek istenildiğinde kurulacak temel hipotez şudur “aynı tür müşteriler, aynı davranışları gösterecektir” [Adriaans ve Zantinge, 1996]. Çok boyutlu veri uzayında bu, aynı tür kayıtlar birbirine yakın olacak demektir. Bundan yola çıkarak, basit ama güçlü bir öğrenme algoritmasından bahsedilebilir. Bu da K-yakın komşuluğudur. K-

yakın komşuluğunda temel düşünce, "komşunun yaptığı gibi yap"tır. Eğer belirli bir kişinin davranışı tahmin edilmek isteniyorsa, veri uzayında o kişiye yakın, örneğin on kişinin davranışlarına bakılır. Bu on kişinin davranışının ortalaması hesaplanır ve bu ortalama belirlenen kişi için tahmin olur.

K-Yakın komşuluğunda, K harfi araştırılan komşuların sayısıdır. 5-yakın komşuluğunda, 5 kişiye ve 1-yakın komşuluğunda sadece bir kişiye bakılır [Han ve Kamber, 2001].

K-Yakın komşuluğu bir öğrenme tekniği değildir. Daha çok bir araştırma yöntemidir. K-Yakın komşuluğu, veri kümesini daha iyi anlamaya yardımcı olur. K-Yakın komşuluğu tekniği ile n tane kayıttan oluşan bir veri kümesinde, her bir kayıt için tahmin yapılmak istendiğinde, her kayıt, diğer kayıtlarla karşılaştırılmak zorundadır. Bu da büyük veri kümelerinde karesel karmaşıklığa sebep olur. Eğer, bir milyon kayıtlı veri tabanında basit bir K-Yakın komşuluğu incelemesi yapılacaksa, bir milyar karşılaştırma yapılması gerekir. Bu, araştırmada sorunlara neden olur. Genelde veri madenciliği algoritmaları n kayıt sayısı kadar karmaşıklığa sahip olmalıdır. Bundan dolayı K-Yakın Komşuluğu tekniği alt-örneklemelerde ya da sınırlı sayıda veri kümesinde kullanılmalıdır.

K-Yakın komşuluğu işleminin uygulanmasında, oldukça fazla sayıda özellik içeren tablolardan dolayı sorunlar ortaya çıkar. Bu özellikler birbirinden bağımsızsa, çok boyutlu uzayda araştırma yapılıyor demektir. Bir milyon veri noktasının eşit dağılımlı bir üç boyutlu uzayı oldukça kalabalık görünmesine rağmen K-Yakın komşuluğu incelemesinden iyi sonuçlar elde edilebilir. Fakat bir milyon veri noktasının eşit dağılımlı bir on iki boyutlu uzayı iyi sonuçlar vermez. Her hangi iki veri noktası arasındaki uzaklık yaklaşık olarak eşit çıkar. Bundan dolayı bir K-Yakın komşuluğu algoritması bu tür veri uzayında iyi sonuç vermez. Bu sorun için iki yaklaşımdan bahsedilebilir. İlki, her bir özelliğin göreceli önemi hesaplanmalıdır yani tablodaki bilgiler, belirli ürünlerde müşterilerin ilgisini tahmin etmeye yarar. İkinci yaklaşım ise veri madenciliğinin karmaşık bir işlem olduğudur. Bazı algoritmalar veri kümesinin bazı bölümlerinde iyi çalışır, fakat diğer algoritmalar için aynı şeyler

geçerli değildir. Veri madenciliği sürekliliği olan bir öğrenme sürecidir. Dolayısıyla veri kümesinin daha iyi anlaşılmasını sağlar.

2.5.5. Görüntüleme teknikleri

Görüntüleme teknikleri, veri madenciliği işleminin başlangıcında, veri kümelerinde bulunan örüntülerin kalitesini belirlemek için kullanılır [Westphal ve Blaxton,1998]. Görüntüleme teknikleri, ileri grafik teknikleri ile kayıtları üç boyutlu ele alarak ilginç seçenekler ortaya çıkarır. Böylece, kullanıcıların üç boyutlu yapılarıdaki karşılıklı etkileşimleri araştırmasını sağlar. Fakat bir çok kullanıcı bu ileri düzey teknikleri uygulamakta zorluk çekebilir. Bu yüzden basit grafiksel gösterim teknikleri de kullanılabilir ve sağlıklı bilgiler edinilebilir. Bu yöntemlerden biri olan serpm diyagramı, başlangıçta uygulanır. İki özellikteki bilginin, bir kartezyen uzayında gösterilmesidir. Veri madenciliği işleminde, serpm diyagramları, veri kümelerinin ilginç alt kümelerini belirlemede kullanılır [Adriaans ve Zantinge, 1996; Zhong ve Zhou, 1999].

2.5.6. Sorgulama araçları ile veri kümesinin ön analizi

Veri madenciliğinde, ilk aşama geleneksel sorgulama araçlarının kullanılarak veri kümesinin yüzeysel bir analizini yapmaktır. Veri kümesine, basit yapılandırılmış sorgulama dilinin (SQL) uygulanmasıyla, sağlıklı bilgiler elde edilir. Fakat ileri örüntü analizi algoritmalarını uygulamadan önce veri kümesinin yapısının bilinmesi gerekir. SQL ile veri kümesinden kolaylıkla elde edilebilen yüzeysel bilgiler ortaya çıkarılır. Geriye kalan gizli kalmış bilgiler ise daha ileri teknikler uygulanarak elde edilir [Maxon, 1996]. Önemli pazarlama araştırmaları yapan büyük kuruluşlar için de bu gizli kalmış bilgiler çok önemlidir.

2.5.7. OLAP

Veri madenciliği, sadece istatistiksel tekniklerin tartışıldığı alan değildir. Aynı zamanda VM amacıyla, veri ambarlamayı (VA) kapsayan çeşitli teknolojileri, teknikleri, çeşitli yazılım paketleri ve dillerin geliştirilmesi ile de ilgilidir. Geleneksel tekniklerin dışında, OLAP'ı içeren çok boyutlu (Multi Dimensional-MD) methodları da kapsar. VM ile ilgili bazı yazılım paketi ve dilleri şunlardır: DBMiner, IBM Intelligent Miner, SAS Enterprise Miner, SGI Mine Set, Clementine, MS/SQL Server 2000, DBMiner, BlueMartini, MineIt, DigiMine, ve VM için MS OLEDB.

Çizelge 2.1. Mevcut VM Teknikleri

TEKNİK	TANIM
Tahminleyici modelleme	Belirli veriler için tahmin değeri
Tanımlayıcı VM	Veri dağılımı, analizi, doğrulama
İlişki Analizi	Özellikler arasındaki ilişkilerin tanımlanması
Sınıflandırma	Hangi veri hangi sınıfa aittir
Kümeleme ve aykırı değer analizi	Benzer özelliklere sahip birimlerin gruplanarak sınıflara ayrılması
Zamansal ve sıralı örüntü analizi	Eğilim ve sapma, sıralı örüntü, periyodik
OLAP	Çok boyutlu veriler içerisinde farklı boyut analizlerinin yapılması
Model görselleştirme	Keşfedilen bilginin daha kolay anlaşılması için görsel öğelerin kullanılması
Keşifçi veri analizi	Tahminlere dayalı veri analizi

Goebel ve Grunwald (1999), çalışmalarında veri madenciliğini, geleneksel istatistik tekniklerinden oldukça fazla etkilenen bir alan olarak tanıtmışlardır. Sınıflandırma, kümeleme, aykırı değer analizi, sıralı örüntüler, zaman serileri analizi, tahmin, regresyon, ilişki çıkarımı ve OLAP'ı kapsayan çok boyutlu yöntemler Çizelge 2.1'de

gösterildiği gibi VM teknikleri olarak kategorize edilmiştir[Goebel ve Grunwald, 1999].

2.6. Veri Madenciliğinin Uygulanabilirliği

Veri madenciliği pratikte çoğunlukla Amerika ve Avrupa'da uygulanmaktadır. American Express ve AT &T gibi büyük kuruluşlar, müşteri dosyalarını analiz etmekte VTBK'ni kullanmaktadır. UK ve BBC sunulan programların analizinde veri madenciliği tekniklerini kullanmaktadır. Avrupa ülkelerinin çoğunda çok sayıda banka ve sigorta şirketleri VTBK ile ilk tecrübelerini edinmişlerdir. VTBK'nin çok problem içerdiği ortaya çıkmıştır. VTBK'nin %80'i veri hazırlama olup geriye kalan %20'si ise madenciliktir. Elde bulunan verilerin, normal veri tabanında rutin olarak yapılan temizleme ve kodlamada kullanılması VTBK işleminin önemli bir bölümüdür. Bu, model oluşturmada daha önemlidir. çünkü doğru veri olmadan doğru bilgiye ulaşamaz.

Veri madenciliğini uygulamak zor değildir. Küçük bir veri kümesinde, veri madenciliği yapılmak istenildiğinde, uzman tarafından sağlanan bir kontrol kümesiyle, kümeleme algoritmasının çalıştırılması gerekir. Bunun için gerekli standart araçlar, hali hazırda piyasadan sağlanabilir. Diğer taraftan büyük bir veri tabanında veri madenciliği yapmak çok daha zordur. Problemlerle karşılaşılabilir ve bundan dolayı projeler başarısızlığa uğrar. Bu problemler:

- 1) İleri görüşün olmaması: Bir dosyanın, gelecekte nelerle karşılaşılacağı ve ne isteneceği düşünülmeden hazırlanması.
- 2) Hazırlanan bütün dosyaların güncel olmaması: Verilerin eksik ya da yanlış olması.
- 3) Kurumlar arasında işbirliği olmaması: Bazı kurumların verilerini vermek istememesi.
- 4) Bilgisayar ortamında veri işlemleri yapan birimler arasında iletişim eksikliğinin olması.

- 5) Resmi ve gizli sınırlamalar ile bazı verilerin gizli olmasından dolayı kullanılamaması.
- 6) Dosyalarla teknik nedenlerden dolayı bağlantı kurulamaması: Veri modellerinin günümüze uygun olmaması ya da hiyerarşik ve ilişkisel veri tabanı arasında farklılık olması.
- 7) Zamanlama sorunu: Dosyaların merkezi olarak toplanması sırasında zaman kaybedilmesi.
- 8) Yorumlama problemi: Veri tabanında keşfedilen ilişkilerin yorumlanamaması.

Yukarıda bahsedilen problemler, sağlıklı ve esnek bir yapıya sahip olan organizasyonlarda çözümsüz değildir [Adriaans ve Zantinge, 1996].

3. VERİ MADENCİLİĞİ VE OLAP

Bağımsız p tane özellik, p boyutlu uzay olarak görülebilir. Buna göre veri kümesinde çok boyutlu analizler ilginç yapılar ortaya çıkarabilir. Kuruluşların cevaplarını aradıkları sorular çok boyutlu analizler gerektirir. Örneğin, magazin dergilerinin yayımcısı, satılan ürün miktarı ile ilgilenmez. Bu sıfır boyutlu bir durumdur. Ama, belirlenmiş bir bölgede, hangi yaş grubu için ne tür magazin dergileri satıldığını bilmek ister. Bu ise dört boyutlu bir sorudur: ürün, satın alan, satılan malın tarihi ve yaş. Bu tür bilgiler çok boyutlu olarak adlandırılırlar ve bu boyutlar arasındaki ilişkiler araştırılmaya çalışılır. Bilgiler daha az boyutlu olduğunda, ilişkileri incelemek kolay olmaz. Bunun için standart veri tabanları çok iyi değildir. Bu tür veri tabanlarında kayıtların takibi çözüm yolu (anahtar) kullanarak yapılır. Çok boyutlu ilişkiler, çok çözüm yolu (anahtar) gerektirir. Boyut sayısı azaldığında çözüm yolu (anahtar) da sınırlanır. Fakat kuruluşların, verilerden elde etmek istedikleri bilgilerin sonu yoktur. Dolayısıyla geniş veri kümelerinde on-line işlem kullanılması tercih edilir. OLAP araçları bu tür problemleri çözmek için geliştirilmiştir. Bu araçlar, verileri, özel çok boyutlu bir formatta hafızada depolar ve veriler güncelleştirilmemiş olsa bile istenilen bilgilere ulaşılır [Adriaans ve Zantinge, 1996].

3.1. Veri Ambarı (VA) Kavramı

Orijinal veri ambarı kavramı IBM tarafından “bilgi deposu” olarak, ilişkileri olmayan sistemlerdeki verilere erişim çözümü sunmak amacıyla geliştirilmiştir. Bilgi deposu, organizasyonlara veri arşivlerine ulaşarak iş avantajı sağlamaya izin vermektedir. Bilgi depolarının bazı problemlere uygulamadaki karmaşıklık ve performans sorunları, bilgi deposu uygulamalarını engellemektedir. Daha sonra veri deposu kavramı ortaya çıkarak bilgi deposu kavramının eksik yönlerini engellemeye çalışmıştır. Bill Inmon tarafından yapılan veri ambarı tanımı genelde kabul edilmiş ve Bill Inmon’a “veri ambarının babası” lakabını kazandırmıştır.

Bill Inmon tarafından 1990'da ortaya atılmış olan veri ambarı terimi şöyle tanımlanmıştır: “Veri ambarı, konuya dayalı, bütünleşik, zamana göre değişen, kalıcı ve yönetimin karar sürecini destekleyen veriler topluluğudur.”

Tanımdaki her bir bileşenle ilgili bilgiler sırasıyla aşağıda detaylandırılmıştır [Kovalerchuk, 2000];

- Konuya dayalı: Veri ambarı organizasyonların temel konuları üzerinde kurulur ve karar destekte kullanılacak verileri depolar.
- Bütünleşik: Kurum genelinde çeşitli veri kaynaklarının biraraya gelmesiyle oluşur. Veri kaynakların gelen veriler örneğin format farklılıklarından dolayı birbirleriyle uyuşmayabilir. Bütünleştirilen veri kaynakları kullanıcılara verilerin birleştirilmiş görünümünü sunmak için birbiriyle uyum sağlamaları sağlanmalıdır.
- Zamana göre değişen: Veri ambarındaki veriler zamanda belirli bir nokta ya da belirli bir zaman aralığı için geçerli ve doğrudur. Veri ambarındaki veriler, verilerin ambara yüklendiği ya da güncellendiği andaki verileri göstermektedir.
- Kalıcı: Veri ambarında veri güncellenmez fakat veritabanından yeni veriler veri tabanına eklenebilir. Veri ambarı, sürekli yeni verileri alır ve eski verilerle bütünleştirir.

Daha sonra çok sayıda veri ambarı tanımı geliştirilmiştir. Özet olarak, veri ambarı veri yönetimi ve veri analiz teknolojisidir.

Son yıllarda veri ambarcılığında yeni bir kavram daha ortaya çıkmıştır web veri evi “data webhouse”. Web veri evi, web üzerinde uygulanan belirli veri havuzuna ihtiyaç duymayan dağıtımli veri ambarıdır [Adriaas ve Zantinge, 1996].

Veri ambarı kavramı, karar vermede kullanılabilecek yapısal kaliteli bilgiye kolay erişimi sağlama ihtiyacından ortaya çıkmıştır. Kuruluşların büyük miktarda verileri olmasına rağmen, ne yazık ki bu verilere erişmek ve kullanmak, veri miktarı arttıkça

daha da zorlaşmaktadır. Bunun sebebi, değişik zamanlarda ve değişik kimseler tarafından geliştirilmiş veri tabanı sistemlerinin ve kütük yapılarının veriyi tutmak için kullanılması, bunun sonucu olarak da çok miktardaki veriye farklı düzlem ve farklı biçimlerden erişme gereksiniminin ortaya çıkmasıdır. Veri ambarındaki veri, daha sonra sorgulama, raporlama ve veri çözümlemede kullanılır.

3.1.1. İşletimsel ve bilişimsel sistemler

Veri ambarının gelişimiyle birlikte ortaya çıkan en önemli noktalardan birisi, bütün kuruluşlarda temel olarak farklı iki çeşit sistem bulunduğu gerçeğidir. Bunlar İşletimsel (*Operational*) sistemler ve Bilişimsel (*Informational*) sistemlerdir.

İşletimsel sistemler, adından da anlaşıldığı gibi kuruluşun günlük işlerini yürütmek için kullanılan sistemlerdir. Sipariş girişi, stok takibi, üretim, muhasebe gibi sistemler kuruluşun temel işlerini yürüten işletimsel sistemlerdir.

Öte yandan, kuruluşun planlaması, ileriye yönelik yönetimsel kararların alınması ve kestirimlerin yapılması için gerekli olan bazı önemli işlevler de söz konusudur. Bu işlevler aynı zamanda hızlı değişen iş dünyasında kuruluşun ayakta kalmasını sağlamak açısından da hayati önem taşır. “Pazarlama planlama”, “Mühendislik planlama” ve “Finansal çözümleme” gibi işlevler, işletimsel sistemler tarafından yerine getirilebilecek işlevler değildir. Bu tip işlevleri yerine getirebilecek sistemler, bilgi tabanlı (*knowledge based*) bilişimsel sistemler olarak adlandırılırlar.

Bilişimsel sistemler, işletimsel sistemlerden farklı amaçlar gütmelerinin yanısıra, bilişimsel veriyi de farklı alanlardan toparlamaları ve büyük miktarda, birbiriyle ilişkili işletimsel veriye gereksinim duymaları açısından da işletimsel sistemlerden farklıdır.

İşletimsel ve bilişimsel sistemlerin temel gereksinimleri birbirinden farklıdır; işletimsel sistemlerde etkinlik ön planda iken, bilişimsel sistemlerde esneklik ve geniş bakış açısı önem taşımaktadır.

3.1.2. Veri ambarı oluşturma nedenleri

İş organizasyonlarında bilgi akış mimarisinde veri ambarları iki amaçla oluşturulmaktadır [Kovalerchuk, 2000]:

1. VA, hareketssel ve organizasyonel görevler arasındaki depo ve analitik stratejik verilerin birikimini sağlar. Bu veriler daha sonra yeniden kullanılmak üzere arşivlenir. VA, verilerin sorgulanabildiği ve analiz yapılabildiği bir depodur.
2. VA'nın pazarda yeni fırsatlar bulmaya, rekabete katkı, yoğun proje çevirimi, iş, envanter, ürün maliyetlerinin azalmasının yanında farklı işlere ait verilerin ilişkilendirilmesi, karar destek ve alınan bilgiye hızlı cevap verebilme gibi birçok katkısı bulunmaktadır.

3.1.3. Veri çözümleme teknikleri

VA, karar verme ve çözümleme amacıyla kullanılacak olan kaliteli veriye kolayca ulaşmak üzere kurulur. Çözümleme için günümüzde en çok kullanılan teknikler, sorgulama ve raporlama (*query and reporting*), çok boyutlu çözümleme (*multidimensional analysis*) ve VM teknikleridir. Bu teknikler, sorgu sonuçlarını formüleştirmek, içeriği farklı bakış açılarına göre çözümlemek ve verinin içeriğindeki örüntüleri ortaya çıkartmak için kullanılmaktadır.

Veri çözümleme teknikleri, seçilen veri modelini ve içeriği etkileyebilmektedir. Örneğin, amaç sadece sorgulama ve raporlama ise veriyi normalize edilmiş bir yapıda tutmak, en hızlı ve kolay erişimi sağlayacaktır. İlgili veri öğelerinin seçilişi ve bu öğelerin belirli ölçütlere göre gruplandırılması ve sonuçların sunuluşu, sorgulama ve raporlama işlemlerinin kapasitesini belirlemektedir. Bu tür işlemlerin, doğrudan tablo taramaları gerektirdiği durumlarda, birim ilişki (*entity-relationship*) modeli en uygun modeldir.

Eğer amaç çok boyutlu veri çözümleme ise, en uygun veri modeli çok boyutlu veri modeli olacaktır. Bu çözümleme yönteminin uygulanabilmesi için veri modelinin, veriye farklı boyut kombinasyonlarında erişimi desteklemesi gerekmektedir. Örneğin, belirli bir günde, belirli bir mağazada, belirli fiyat aralığında, belirli bir üründen kaç adet satıldığı öğrenilmek istenebilir. Başka bir çözümlemede ise belirli bir ürünü, belirli bir günde, belirli bir fiyat aralığında satan mağazaların sayısı araştırılıyor olabilir. Bu iki sorgu, benzer bilgileri istemesine rağmen, bir tanesi olaya ürün bakış açısından yaklaşırken, diğeri mağaza açısından yaklaşmaktadır.

Çok boyutlu çözümleme, verinin birçok farklı bakış açısından ve boyuttan kolayca görülebilmesini sağlayan bir veri modeli gerektirmektedir. Normalize edilmiş bir veri modeli kullanılması durumunda, farklı boyutlardaki verileri tutan tablolar arasında birçok birleştirme (join) işlemi yapılacağından, başarımlar gözle görülür oranda kötü etkilenecektir.

VM yaklaşımında, kullanılacak VM sorgusuna ve uygulamaya göre, verinin modeli değişmektedir. Birçok uygulama, veri ambarının değişen gereksinimlerini karşılayacak olan en uygun veri modelini bulmak için birden fazla veri modeli kullanabilir.

3.1.4. Veri ambarının üretilmesi

VA'nın üretilmesi, işletimsel ve dış sistemlerdeki verilerin veri ambarına aktarılması ile gerçekleştirilmiştir ve oluşturulan veri ambarı sayesinde veri madenciliği sorgularının yapılacağı veri kümesi oluşturulmuştur. İşletimsel ve dış sistemlerden alınan veri, veri ambarında kullanılacak uygun biçime dönüştürüldükten sonra veri ambarına yüklenmektedir.

Veri yakalama

Veri yakalama işlemi, işletimsel sistemler ve diğer dış kaynaklardaki verilerin toplanması işlemidir. Yakalama işlemindeki veri kaynağı, kütük biçimleri ile hem

ilişkisel hem de ilişkisel olmayan Veri Tabanı Yönetim Sistemleri(VTYS)'ni içermektedir. Veri, görüntü kopyaları, değiştirilmiş veri kütüğü ya da tabloları, veri tabanı günlükleri, ileti kütükleri ve olay tutanakları gibi birçok farklı türdeki kütükten yakalanabilir. Verinin ne tür bir kütükten yakalandığı, veri yakalamada kullanılan tekniğe bağlıdır. Bu teknikler;

- Kaynak veri çıkartma (*source data extraction*),
- VTYS günlük yakalama (*DBMS log capture*),
- Tetiklenmiş yakalama (*triggered capture*),
- Uygulama yardımıyla yakalama (*application-assisted capture*),
- Zaman-damgası-tabanlı yakalama (*time-stamp-based capture*)
- Karşılaştırmalı yakalama (*comparison capture*)

Kaynak veri çıkartma, kaynak verinin belirli bir zamandaki durağan bir görüntüsünü (*snapshot*) verir. Kaynak veri çıkartma işlemi sonucunda kütükler, tablolar ya da resimler oluşabilir.

Günlük yakalama, verinin veri tabanı yönetim sisteminden günlük sisteminden yakalanabilmesini sağlar. Veritabanı ya da veritabanına erişen işletimsel sistemler üzerinde çok az bir etkisi vardır. Bu teknik, sadece ilgilenilen verinin yakalanabilmesini sağlamak üzere, günlük kayıtlarının nasıl bir biçimde tutulduğunun bilinmesini ve karmaşık programlar geliştirilmesini gerektirmektedir.

Tetikleyiciler, birçok veri tabanı yönetim sistemi tarafından desteklenen ve veri tabanında gerçekleşen belirli bir olayın sonucunda SQL deyimlerinin ya da daha karmaşık programların işletilmesini sağlayan yordamlardır. Tetikleyiciler sayesinde herhangi bir çeşit yakalama işlemi gerçekleştirilebilir. Tetikleyici, belirlenen olayın gerçekleştiğini fark eder ve kullanıcının geliştirdiği yordamı çalıştırır. Bu teknik kullanılırken, belli zamanlarla işletimsel veriler yakalanacak şekilde sistem tasarlanmıştır.

Uygulama yardımıyla yakalama, var olan işletimsel sistem uygulamalarındaki programlama mantığına dayanır. Bu, bütün test ve yönetim sorumluluklarının uygulama programcısında olduğu anlamına gelmektedir. Geçerli bir teknik olmasına rağmen, uygulama yardımıyla yakalama tekniğinin kullanılması için kurumların kendi programlarını geliştirmeleri yerine, bu amaç için geliştirilmiş hazır programları kullanmaları daha yerinde olabilir.

Zaman-damgası-tabanlı yakalama, bir kaydın son yakalamadan bu yana değişip değişmediğini belirlemek üzere kullanılan belirli zaman damgasına dayanan basit bir tekniktir. Eğer bir kayıt değişmişse ya da yeni bir kayıt eklenmişse, bu kayıt işlenmek üzere bir kütüğe ya da tabloya alınır.

Kütük karşılaştırma, uzun yıllardır kullanılan bir tekniktir. Çok etkin bir yöntem olmamasına rağmen, anlaşılması ve uygulanması kolaydır. Veri yakalama anında veri kaynağının anlık görüntüsünün (snapshot) alınması ve bu görüntünün daha önce alınan görüntüyle karşılaştırılması şeklinde gerçekleştirilir. Belirlenen herhangi bir değişiklik, daha sonra işlenir.

Veri dönüştürme

Dönüştürme işlemi, yakalanan kaynak veriyi veri ambarına yüklenebileceği uygun biçim ve yapıya dönüştürür. Kaynak verinin dönüştürülmesi için kullanılan eşleştirme ölçütleri yakalanarak üstveri şeklinde saklanır. Bu üstveri, verinin veri ambarına yüklenmesinden önce yapılması gereken değişiklikleri tanımlar. Dönüştürme işlemi, kaynak verideki anormallikleri ortadan kaldırarak hedef veri ambarı için yüksek kaliteli verinin elde edilmesini sağlar. Verinin dönüştürülmesi, kayıt ya da nitelik düzeyinde gerçekleştirilebilir. Temel dönüştürme teknikleri, yapısal dönüştürme (structural transformation), içerik dönüştürme (content transformation) ve işlevsel dönüştürme (functional transformation) teknikleridir.

Yapısal dönüştürme, kaynak kayıtların yapısını hedef veri tabanına göre değiştirme işlemidir. Bu teknikte veri, kayıt düzeyinde dönüştürülür.

İçerik dönüştürme kayıtlardaki verilerin değerlerini değiştirme işlemidir. Bu işlem, veriyi nitelik düzeyinde dönüştürmektedir. İçerik dönüştürme, algoritmalar ya da veri dönüştürme tabloları kullanılarak gerçekleştirilebilir.

İşlevsel dönüştürme, kaynak kayıtlardaki verilere dayanarak, hedef kayıtlarda yeni veri değerleri yaratılması işlemidir. Bu teknikte de veri, nitelik düzeyinde dönüştürülür.

Uygulamaya koyma

Uygulamaya koyma işlemi, dönüştürme işleminde yaratılan kütük ve tabloları işleyerek, bunları ilgili veri ambarı ya da veri çarşısında (data mart) uygulamaya koyma işlemidir. Bu işlem için 4 temel teknik vardır: yükleme (load), ekleme (append), yapıcı birleştirme (constructive merge) ve yıkıcı birleştirme (destructive merge).

- Yükleme, hedef veri ambarı tablolarını, dönüştürme işleminde yaratılan tablolarla değiştirir. Eğer veri ambarında bu tablolar yoksa, tablolar yaratılacaktır.
- Ekleme, varolan tabloların sonuna dönüştürme sonucunda ortaya çıkan kütük ya da tablodaki yeni verileri ekler.
- Yapıcı birleştirme, var olan hedef tablolara yeni verileri ekleyerek, durumu değişen kayıttaki bir zaman değerini günceller.
- Yıkıcı birleştirme ise, yeni verileri var olan kayıtların üzerine yazar.

3.2. Veri Ambarı Mimarisinde Bileşenler

3.2.1. İşlevsel veri

Veri ambarı için veri kaynağı:

- Ana işlevsel veri, birinci kuşak hiyerarşi ve ağ veritabanlarından elde edilir. Bu sistemlerden elde edilen ortak işlevsel verinin büyüklüğü tahmin edilebilir.
- Bölümlere ait veriler özel dosya sistemlerinden elde edilir.
- Özel veriler iş istasyonları ve özel sunuculardan elde edilir.
- Dış sistemler, internet gibi herkesin erişebileceği veritabanları ya da tedarikçi ve müşteriler arasında oluşturulan veritabanlarıdır.

3.2.2. İşlevsel verideposu (Operational Datastore-ODS)

ODS, analiz edilecek anlık ve tümleşik işlevsel verinin depolandığı yerdir. Yapısal olarak ve verinin sağlanması bakımından veri ambarına benzer, fakat daha basittir. ODS kullanıcılara, veri ambarının karar destek fonksiyonlarından uzak kalındığında ilişkisel veritabanı kullanım kolaylığı sağlar. ODS oluşturmak, veri ambarı oluşturmayı kolaylaştıran bir adımdır. Çünkü ODS sistemden çekilmiş ya da silinmiş veriyi sağlayabilir.

3.2.3. Yük yöneticisi

Yük yöneticisi, veri ambarına veri yükleme operasyonlarını yapar. Veri, veri kaynağından ya da ODS'den doğrudan çekilebilir. Yük yöneticisi tarafından yapılan işlem, ambara giriş için hazırlanmış verinin basit taşınmasıdır. Bu bileşenin karmaşıklığı ve büyüklüğü veri ambarları arasında çeşitlilik göstermektedir.

3.2.4. Ambar yöneticisi

Ambar yöneticisi, verinin ambar içindeki bütün operasyonlarını yönetir. Ambar yöneticisi tarafından yapılan işlemler:

- Tutarlılık sağlamak için veri analizi
- Veri kaynaklarını geçici depodan veri ambarı tablolarına taşımak
- Temel tablolarda indeksler oluşturmak

- Gerekli olduğunda normalleştirmeler yapmak
- Gerekli olduğunda birleştirmeler yapmak
- Verileri yedeklemek ve arşivlemek

3.2.5. Sorgu yöneticisi

Sorgu yöneticisi, kullanıcı sorgularını yönetir. Bu bileşen tarafından yapılan işlemler, uygun sorguları uygun tablolara yönleltmek ve sorguların yapımını programlamaktır.

3.2.6. Ayrıntılı veri

Veri ambarının bu alanında, veritabanı şemasındaki ayrıntılı veriler saklanır.

3.2.7. Az ve çok özetlenmiş veri

Veri ambarının bu alanında, veri ambarı yöneticisi tarafından üretilmiş, önceden özetlenmiş veriler depolanır. Bilgi özetinin amacı, sorgu performansını hızlandırmaktır. Özet veri, ambara yeni bilgi eklendikçe devamlı olarak güncellenir.

3.2.8. Arşivlenmiş/Yedeklenmiş veri

Veri ambarının bu alanında arşivleme ve yedekleme amacıyla detaylandırılmış ve özetlenmiş veri depolanır. Veri, manyetik bant ya da optik disk gibi depo arşivlerine transfer edilir.

3.2.9. Meta-Veri

Veri ambarının en önemli bileşenlerinden biri meta-veridir. Meta-veri VA’nda verilerin tanımlandığı kısımdır. Meta-veri “veri hakkında veri” anlamındadır. Meta-veri her veri elemanının anlamını, hangi elemanların hangileriyle nasıl ilişkili olduğunu ve kaynak verisi ile erişilecek veri gibi bilgileri içermektedir. Veri

ambarındaki veriler, VA yöneticisinin kullandığı teknik veriler ve VA kullanıcılarının kullandığı iş verileri olarak ikiye ayrılır:

Teknik veriler: Operasyonel veritabanı tanımlarını ve veri ambarı tanımlarını içerir. Bu iki tanım veya şema veri ambarını çalıştırılabilmesini sağlayan veri taşıma operasyonlarını içerir. Bu bilgiler, veri ambarı yöneticisine veri ambarında birbiriyle ilişkili verileri göstererek yardımcı olan bilgilerdir.

İş verileri: Kullanıcılara yardım eder. Kullanıcıların veritabanı oluşturan veriler dışındaki veri ambarında bulunan bilgilere ulaşmalarına yardımcı olur. Ayrıca veri ambarına verinin ne zaman ve nereden geldiği gibi bilgilere ulaşılmasını sağlar.

3.2.10. Son kullanıcı erişim araçları

VA'nın ana amacı, stratejik karar almak için kullanıcılara bilgi sağlamaktır. Kullanıcılar, son kullanıcı erişim araçlarını kullanarak ambarından faydalanırlar. Veri ambarı, özel ve rutin analizleri etkili olarak destekliyor olmalıdır.

Bu araçlar beş ana gruba ayrılabilir:

- i) Raporlama ve sorgulama araçları
- ii) Uygulama geliştirme araçları
- iii) Yönetim bilgi sistemi araçları
- iv) OLAP araçları
- v) Veri madenciliği araçları

3.3. Veri Ambarında Veri Akışları

Bu kısımda VA'ndaki veri akışları ve bu akışların fonksiyonları belirtilmiştir. VA beş ana veri akışını yönetmeye odaklanmaktadır.

VA'daki veri akışları ve bu akışların uyguladığı işlemler aşağıdaki gibidir:

- İçeriye akış; özetleme, temizleme ve veri kaynağından yükleme.
- Yukarıya akış; veriyi özetleyerek, paketleyerek ve dağıtarak veriye değer kazandırmak.
- Aşağıya akış; ambarda veriyi arşivlemek ve yedeklemek.
- Dışarıya akış; son kullanıcı için veriyi kullanılabilir yapmak.
- Meta akış; meta- veri'yi yönetmek.

3.4. Çok Boyutlu Veritabanı Tasarım Şemaları

Veri ambarı uygulama şemasının tasarımı, veritabanı tasarımında özel bir konudur. Birim-ilişki veri modeli, birim ve ilişki kümelerini ve bunlar arasındaki ilişkileri bir şema halinde gösteren ilişkisel veritabanlarının tasarımında sıklıkla kullanılan modellerdir. Böyle bir veri modeli, online veri işleme süreci için uygundur. Bununla birlikte bir veri ambarı, özet şekilde sunulan, online veri analizini kolaylaştıran, amaca yönelik bir şemaya ihtiyaç duyar. İki konunun özel önemi olduğu varsayılır:

- Verinin geniş hacmi
- Yeniden elde etme performansında tatmin edici seviyelerin sağlanması sorunu

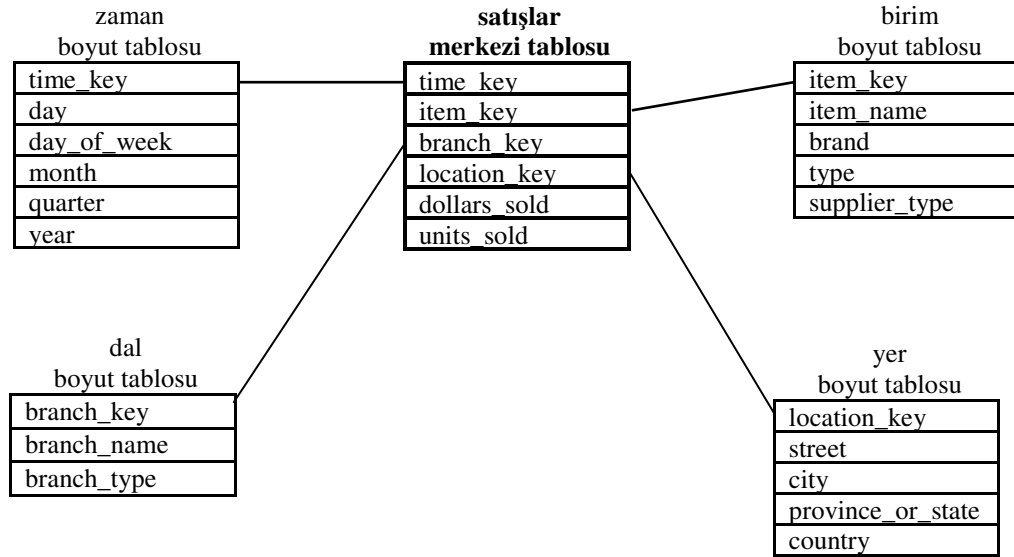
Performansta tatmin edici seviyenin sağlanması, amarlama uygulamalarında uzmanlaşmış fiziksel şemaların tasarlanmasıdır.

Veri ambarları için en popüler veri modeli, çok boyutlu modellerdir. Amarlama şeması için yaygın olan üç model tasarımı vardır [Kovalerchuk, 2000; Han ve Kamber, 2001].

- 1) Yıldız şema (Star Schema)
- 2) Kar tanesi şeması (Snowflake Schema)
- 3) Takımyıldızı olgusu şema (Fact Constellations)

3.4.1. Yıldız şema

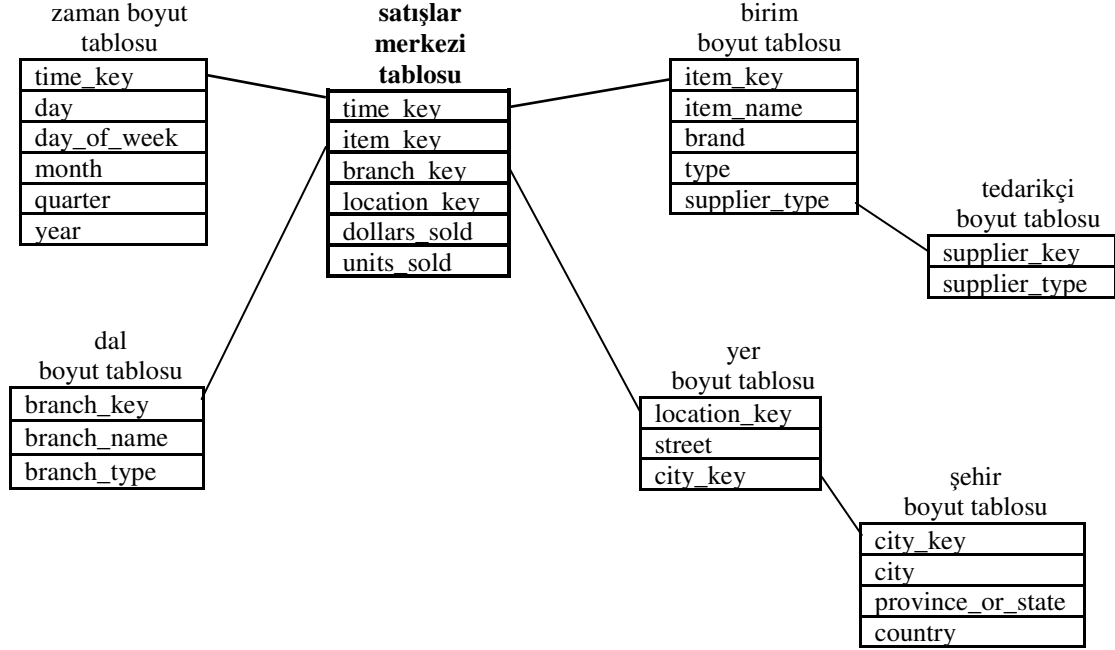
Bu tip şemalar, (1) gereksiz bilgiyi içermeyen, büyük boyutlu veri hacmi bulunan merkezi tablo(fact table), ve (2) her boyut için bir tane olmak üzere daha küçük yardımcı boyut tabloları (dimension table) kümesi içeren, VA için sıklıkla kullanılan modelleme şekilleridir. Yıldız şema, merkezinde ‘olgusal veri’yi ve bunu çevreleyen referans tablolardan oluşan fiziksel şemadır. Merkezi olgusal veri, organizasyonel aktivitelerden elde edilen verilerin depolanması için tasarlanmıştır. Referans tablolar, yeniden elde etme performansını geliştirmek için bir takım diğer tabloların normalleştirilmiş versiyonlarıdır. Şekil 3.1’te satış verileri üzerinden hazırlanmış yıldız şeması gösterilmektedir [Kovalerchuk, 2000; Han ve Kamber, 2001].



Şekil 3. 1. Yıldız şema

3.4.2. Kar tanesi şeması

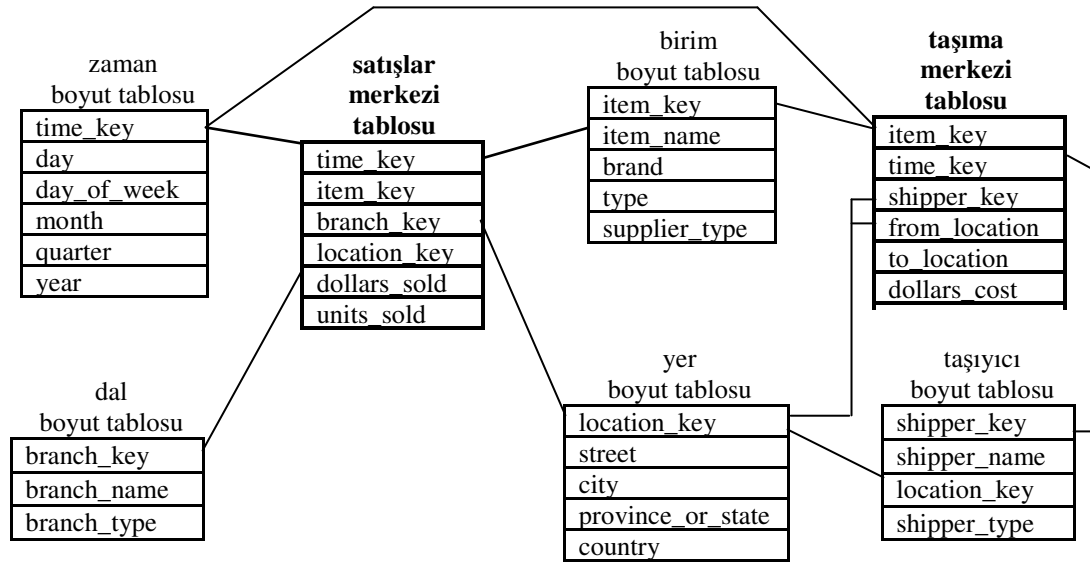
Kar tanesi şeması, yıldız şemasının bir çeşididir. Kar tanesi şemasında, her boyutun kendine ait boyutları vardır. Bu da referans tabloların normalize edilmesi demektir. Sonuçta ortaya çıkan şema, kar tanesinin yapısına benzer bir şekil almaktadır. Aşağıdaki Şekil 3.2’de bir satış verileri üzerinde uygulanmış kar tanesi şeması gösterilmektedir [Kovalerchuk, 2000; Han ve Kamber, 2001].



Şekil 3. 2. Kar tanesi şeması

3.4.3. Takımyıldızı olgusu şema

Yıldız ve kar tanesi şeması arasında yer alır. Bazı karmaşık uygulamalar, boyut tablolarını paylaşan birden fazla merkezi tabloya (fact table) ihtiyaç duyabilir. Takımyıldızı şemada bazı referans tablolar normalize edilmiş, bazıları da denormalize edilmiştir. Bu tip şemalar, birden fazla yıldız şemanın bir araya gelmiş haline benzemektedir. Bu nedenle bu tip şemalara “galaksi şema” ya da “takım yıldızı olgusu” adı verilmektedir. Aşağıdaki Şekil 3.3’de satış verileri örneği üzerinden hazırlanmış takımyıldızı olgusu şema gösterilmiştir [Kovalerchuk, 2000; Han ve Kamber, 2001].



Şekil 3. 3. Takımyıldız olgusu şema

3.5. Veri Tabanı İle Veri Ambarı Karşılaştırması

Veri tabanları, önceki konularda bahsi geçen yapılarda oluşturulduktan sonra, organizasyondaki birimler ihtiyaç duyduğu kısmı veritabanından çekerek kendisine ait bir veri ambarı oluşturur. Veri ambarları veritabanlarına göre çok daha esnek hareket kabiliyetine sahiptir.

Çizelge 3. 1. Veri tabanı ile veri ambarının karşılaştırılması

	VERİ TABANI	VERİ AMBARI
Konu Odağı	Veri, bir fonksiyonla veya işlem yönlendirmesi ile saklanır.	Veriler konu odaklıdır, fonksiyonlara ve süreçlere çoklu bakışı yansıtır.
Zaman Odağı	Verilerdeki güncel hareket ve olaylar kaydedilir.	Veri birkaç zaman periyodunu içeren tarihsel yönlendirmeye sahiptir.
Uçuculuk	Veri güncellemeler rutindir ve sık sık olabilir.	Veriler değişmez (örneğin, detaylar salt okunurdur). Fakat, yeni zaman periyotlarını ve ilgi alanlarını yansıtan, yeni veriler veri ambarlarına eklenebilir.

3.6. Çevrimiçi Analitik İşleme (OLAP)

İlişkisel veri tabanlarının yaygınlığı ve sonrasında ortaya çıkan Veri Ambarlarının gelişmesi ile beraber, verilere daha hızlı şekilde erişme ve çok boyutlu analiz ihtiyaçları, bilim adamlarını ve yazılım şirketlerini, daha farklı yapılar geliştirmeye itmiştir.

Bu amaçla geliştirilen bir teknoloji olan Çevrimiçi Analitik İşleme (OLAP), ilişkisel veri tabanları gibi, bilimsel temeller üzerine değil, OLAP ürünleri üreten firmaların desteğinde çıkan bir teknoloji olmuştur. Bu nedenle, veri tabanları, ilişkisel veri tabanları ve hatta veri ambarları üzerine birçok akademik yazı bulunmasına rağmen, OLAP üzerine genellikle sadece ürün dokümanları ve şirketlerin tanıtım yazıları bulunabilmektedir.

OLAP teriminin ilk olarak ortaya çıkışı ise, 1993 yılında, Dr. E.F.Codd 'un ortaya koyduğu kurallar çerçevesinde olmuştur. Bu yazı, OLAP için bir temel oluştursa da, kimi çevrelere göre, o yıllarda Arbor Software (Şimdiki Hyperion Solutions) için bir makale olmaktan öteye gidememiştir.

Bu yazıya göre, bir veri yapısının OLAP olarak nitelendirilebilmesi için 12 kural belirlenmiştir. Bu kurallar sırası ile:

- Çok boyutlu inceleme özelliğine sahip olması,
- Şeffaflık,
- Erişilebilirlik,
- Her seviyede sorgulama için aynı performansı gösterebilme özelliği,
- İstemci-Sunucu yapısında olması,
- Sınırsız şekilde çapraz raporlama olanağının olması,
- En alt seviyedeki verilerin otomatik olarak ayarlanması,
- Her şarta uygun boyutlandırılabilirlik,
- Çok kullanıcı desteğinin olması,

- Her seviyede verilerin değiştirilebilir olması,
- Esnek raporlama özelliği,
- Boyut ve gruplamalarda sınır olmaması.

şeklindedir.

3.6.1. OLAP'ın ticari boyutu

Kullanılan sektörler/alanlar

OLAP, yöneticiler ve analistlerin, verilere çok hızlı şekilde, farklı açılardan bakabilmelerini sağlayan bir yapıdır. “Kim?” ve “Ne Zaman?” sorularından başka, “Neden?” ve “Eğer şu olursa...” sorularının da yanıtını verir. (Ör : Eğer şeker fiyatları 5% ve taşıma maliyetleri 10% düşerse, yıllık ve çeyrekler bazında kârlılık ne olur gibi.)

Akıllı raporlama araçları sayesinde, neden sorularının cevapları da kolaylıkla alınabilmektedir. Genel eğilimden farklılık gösteren, uç değerler yaratan elemanları birçok analiz aracı, sayısal detaylara girmeden, sadece renklerle bile görüntüleyebilmektedir.

OLAP'ı sadece büyük özet tablolar gibi yorumlamak pek doğru değildir. Excel kullanıcılarının yakından tanıyacakları Pivot tabloların, çok gelişmiş ve hızlı bir hali olarak göz önüne getirmek daha doğru olacaktır. Tasarlanan bir OLAP yapısının, hiyerarşilerini ve boyutlarını görmek mümkün olsa da, verileri nasıl tuttuğunu, iki boyutlu grafikler olarak göstermek mümkün değildir, ancak iç içe geçmiş küpler olarak yorumlanabilir. Bu nedenle OLAP yapılarına, “küp” adı verilmektedir.

Bir veri ambarınızın olması, OLAP'a ihtiyacınız olmadığı anlamına gelmez. Veri Ambarları ve OLAP birbirlerini tamamlarlar. Veri Ambarı, verileri uygun şekilde

tutmaya ve kontrol etmeye yarar. OLAP ise, ambar verilerini stratejik bilgilere dönüştürür.

Bir şirket yapısı içerisinde, bölümler bazında inceleyecek olursak; Pazarlama bölümlerinde OLAP'ın en yaygın kullanım alanları, pazar araştırmalarında, satış tahminleri, promosyon ve kampanya analizleri, müşteri analizleri ve pazar/müşteri segmentasyonlarıdır. VM sonuçlarının değerlendirilmesi ve demografik özellikler bazında incelenmesi seviyesinde de olmazsa-olmaz araçlardan biri olarak yer almaktadır.

Üretim ile ilgili uygulamaları ise en yoğun olarak üretim planlama ve hata analizleridir. Özellikle senaryo geliştirmekte ve farklı ürün tipleri ile çalışılan yapılarda, çok boyutlu düşünme imkanı sayesinde maliyetler ve fiyatlamalar, kolaylıkla çıkarılabilmektedir.

Finans bölümleri ise OLAP'ı bütçeleme, Faaliyet Tabanlı Maliyetlendirme (Activity-Based Costing), finansal performans analizleri ve finansal modelleme amaçları ile kullanabilir. Özellikle konsolidasyon konusunda yaratılacak modeller, çok büyük kolaylıklar sağlamaktadırlar. Strateji belirleme, satış analizleri ve gelecek tahminleri ise, satış bölümlerindeki OLAP uygulamalarıdır.

OLAP Yazılımlarının Pazardaki Yeri

OLAP'a yönelik pazar araştırmasını içeren güncel çalışmalardan bir tanesi 2006 yılında gerçekleştirilmiştir [Pendse, 2006]. Gerçekleştirilen anket çalışmasına dünyanın 78 ülkesindeki 31 farklı endüstriden 1700'e yakın kullanıcı katılmıştır. Değerlendirmede 13 OLAP yazılımı; Microsoft Analysis Services, Applix TM1, Cognos Analysis (Cognos 8 or PowerPlay), MicroStrategy, MIS, Essbase, SAP BW, Business Objects WebIntelligence or Desktop Intelligence, Outlook Soft, BOARD Management Intelligence Toolkit, MIK OLAP, Oracle Discoverer and Oracle OLAP servers, dikkate alınmıştır. Anket sonuçları aşağıdaki başlıklar altında değerlendirilmiştir.

OLAP Yazılımları - Örneklem:

OLAP farklı iş kollarında yaygın olarak kullanılmaktadır. Bununla birlikte bazı sektörlerdeki kullanımı diğer sektörlerle göre daha üst düzeydedir. Bu alanlar; sigorta, banka ve perakende sektörleridir. Şekil 3.4'de ürüne göre satış yüzdeleri görülmektedir[Pendse, 2006].

	Tümü	BOARD	BusinessObjects	Cognos Analysis	Essbase	Microsoft AS	MicroStrategy	MIK OLAP	MIS	Oracle Discoverer	Oracle OLAP Servers	SAP BW	TMI	MOLAP	ROLAP
Sigorta ve Finans Hizmeti	9,1	4,8	12	7,4	11	11	8,4	11	6,7	11	5,9	4,2	9,4	9,5	7,9
Bankacılık	8,4	1,6	12	8,5	12	7,8	14	2,2	3,7	0,0	14	4,2	8,9	8,0	9,4
Parekendecilik	6,7	0,0	4,1	4,2	9,7	5,5	19	6,5	2,2	5,4	12	4,2	5,7	5,4	11
İmalat - Hızlı Tüketim Malları	5,5	11	1,4	6,3	8,9	7,8	4,2	2,2	0,0	2,7	12	6,3	2,6	6,0	4,0
Sağlık Hizmeti	5,1	11	2,7	8,5	1,6	4,8	7,2	2,2	3,0	5,4	9,8	4,2	3,6	5,1	5,2
İmalat - Endüstriyel Ürünler	4,9	14	4,1	4,2	4,8	4,5	1,2	6,5	7,4	5,4	5,9	4,2	3,1	5,2	4,0
Telekomünikasyon	4,3	0,0	8,1	2,1	5,6	4,5	4,8	2,2	3,7	2,7	9,8	6,3	2,6	3,7	6,2
Bilişim Teknolojileri(BT) - Servis	4,1	1,6	9,5	9,5	0,8	5,0	3,0	2,2	0,7	2,7	2,0	6,3	1,0	3,8	5,2
Otomotiv	3,9	1,6	1,4	4,8	2,4	2,0	1,2	13	12	5,4	2,0	3,2	5,2	4,4	2,2
İmalat - Süreç	3,6	9,5	0,0	2,1	2,4	4,3	0,0	0,0	5,2	14	0,0	6,3	4,7	3,8	3,0
Eczacılık	3,6	4,8	2,7	6,3	2,4	2,5	4,2	2,2	5,9	0,0	3,9	3,2	3,6	3,8	3,0
Yağ, Gaz ve Kimyasallar	3,4	4,8	4,1	1,6	2,4	3,8	0,6	2,2	5,9	0,0	0,0	6,3	6,8	3,7	2,7
Toptancılık	3,2	3,2	1,4	2,1	8,1	3,8	1,8	4,3	3,0	2,7	0,0	6,3	1,6	3,4	2,7
Danışmanlık ve İlgili Hizmetler	3,1	1,6	8,1	2,1	2,4	4,3	1,8	0,0	3,0	0,0	2,0	1,1	4,7	3,3	2,7
Hükümet	3,0	3,2	5,4	5,8	0,8	2,0	5,4	0,0	1,5	16	0,0	3,2	1,0	2,2	5,4
Taşımacılık	2,9	0,0	1,4	2,1	4,0	2,0	2,4	4,3	5,9	2,7	0,0	2,1	5,7	3,1	2,2
Yiyecek ve İçecek	2,8	6,3	1,4	2,1	4,0	3,5	0,0	11	2,2	0,0	3,9	3,2	1,6	3,3	1,2
Enerji ve Yan Sanayii	2,5	0,0	0,0	3,2	2,4	2,0	2,4	0,0	4,4	5,4	3,9	2,1	3,1	2,5	2,2
Medya ve Basım-Yayım	2,0	1,6	4,1	1,1	0,8	1,0	2,4	6,5	2,2	2,7	0,0	4,2	2,1	1,6	3,5
BT - Yazılım	2,0	0,0	1,4	3,2	4,0	2,8	3,0	0,0	0,0	0,0	3,9	2,1	0,0	2,0	2,0
İnşaat ve Mimarlık	1,7	4,8	0,0	1,1	0,0	1,5	0,6	6,5	3,7	0,0	2,0	2,1	2,1	2,0	0,7
Eğitim	1,5	3,2	4,1	1,6	0,0	0,8	3,0	0,0	1,5	5,4	2,0	1,1	0,5	1,0	3,0
Uzay	1,1	0,0	1,4	1,1	2,4	0,0	0,0	0,0	0,7	0,0	3,9	4,2	2,1	1,0	1,5
Turizm ve Seyahat	1,0	0,0	0,0	2,1	0,8	1,3	1,2	0,0	0,7	0,0	0,0	0,0	2,1	1,2	0,5
Reklam ve Pazarlama	0,9	0,0	0,0	1,1	0,0	2,0	0,6	0,0	1,5	0,0	0,0	0,0	1,0	1,1	0,2
Eğlence	0,9	0,0	2,7	1,1	0,0	0,8	1,8	0,0	0,0	0,0	0,0	0,0	1,6	0,7	1,5
Tarım	0,7	6,3	1,4	0,5	0,0	0,5	0,6	0,0	0,0	2,7	0,0	0,0	0,5	0,7	0,7
BT - Donanım	0,5	0,0	0,0	1,1	0,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	1,0	0,7	0,0
Askeri	0,5	0,0	2,7	0,0	0,0	0,3	0,6	2,2	0,0	2,7	0,0	2,1	0,0	0,2	1,5
Madencilik	0,5	0,0	0,0	0,0	0,0	0,5	0,0	0,0	0,7	0,0	0,0	2,1	1,6	0,5	0,5
Dernek ve Hayır Kurumları	0,5	0,0	0,0	0,5	0,0	1,5	0,0	0,0	0,0	0,0	0,0	0,0	0,5	0,7	0,0
Hava Yolları	0,3	0,0	0,0	0,0	0,0	0,5	0,0	0,0	0,0	0,0	0,0	1,1	0,0	0,3	0,2
Diğer	5,8	4,8	2,7	2,6	5,6	4,5	4,2	13	13	5,4	2,0	4,2	9,9	6,3	4,2

Şekil 3. 4. Ürüne göre satış yüzdeleri

OLAP Yazılımları - Fayda Analizi

Fiyat ve performans ile karşılaştırıldığında, fonksiyonelliği OLAP yazılımının seçiminde en önemli kriter olarak göze çarpmaktadır.

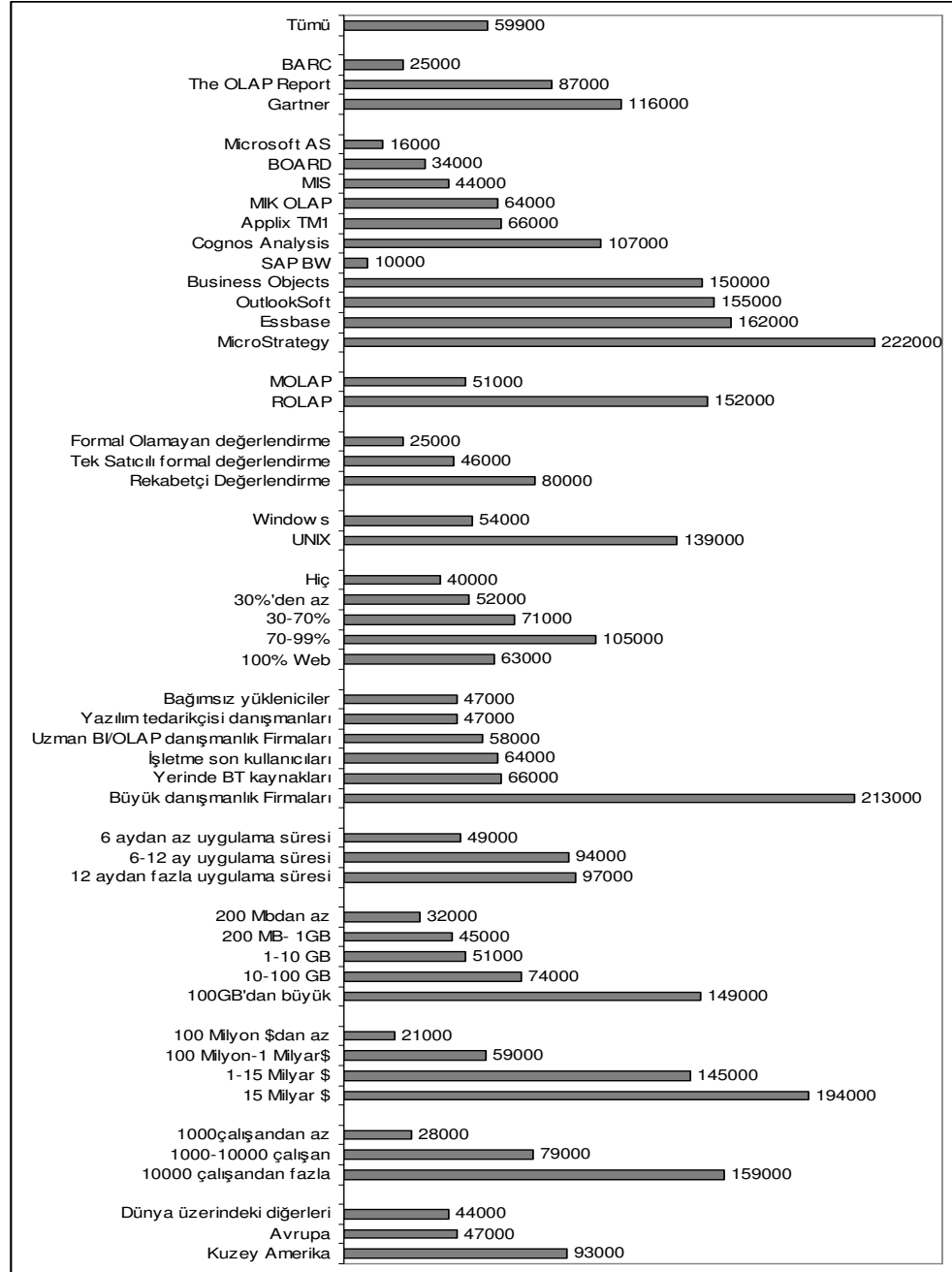
Ürün fonksiyonelliği ve özellikleri her ne kadar önemli olsa da, yazılımın genel başarısı işletmeye kattığı fayda/değer ile ölçülmektedir. İş Fayda İndeksi (Business Benefits Index), bu tür bir ölçektir. Şekil 3.5’de ürün bazında fayda ve hedef skorlar görülmektedir. BBI, yazılımın faydasının 9 kriterden elde edilen sonuçların ağırlığını dikkate alarak hesaplar. BBI indeksinin büyüklüğü yazılımın işletmeye sağladığı katkının göstergesidir.

	Tümü	Microsoft AS	BOARD	Applix TM1	Micro Strategy	MIS	Cognos Analysis	Essbase	Outlook Soft	Business Objects	SAP BW
BBI	4.208	4.54	4.44	4.41	4.36	4.34	4.22	4.05	3.25	3.18	3.06
Hızlı ve daha doğru raporlama	7.568	7.57	7.96	8.39	7.53	8.52	7.39	7.91	7.25	5.42	6.16
Daha iyi karar alma	6.178	6.53	6.59	6.73	5.89	6.94	5.79	6.20	5.93	4.35	5.81
Müşteri memnuniyetini artırma	4.022	4.47	3.82	3.85	4.97	3.48	4.52	4.22	2.82	2.95	2.48
İnsan gücünden tasarruf	3.561	3.47	3.32	4.01	3.51	4.34	3.33	4.09	3.00	2.71	2.09
Daha iyi satış ve pazarlama analizleri ile gelirleri artırma	3.167	4.12	4.26	2.35	4.13	2.17	3.66	2.50	0.48	3.02	1.73
IT olmayan maliyetlerden tasarruf	3.102	3.37	3.76	3.19	3.08	2.49	3.52	2.71	2.53	2.59	1.86
Dışa bağımlı IT maliyetlerinde azalma	2.840	3.43	2.35	3.18	2.46	2.38	2.95	2.30	0.84	1.74	2.06
IS personelinden tasarruf	2.707	2.86	2.75	2.67	3.25	1.99	2.79	2.46	1.92	2.51	1.52
Diğer	2.948	3.33	10.0	2.18	2.64	4.08	1.67	2.22	8.33	2.00	0.67
Hedeflenen başarı	6.765	6.75	6.58	7.55	6.59	7.72	6.20	7.36	5.81	5.64	6.54

Şekil 3. 5. Ürün bazında fayda ve hedef skorlar[Pendse, 2006]

OLAP Yazılımları- Lisans Ücretleri

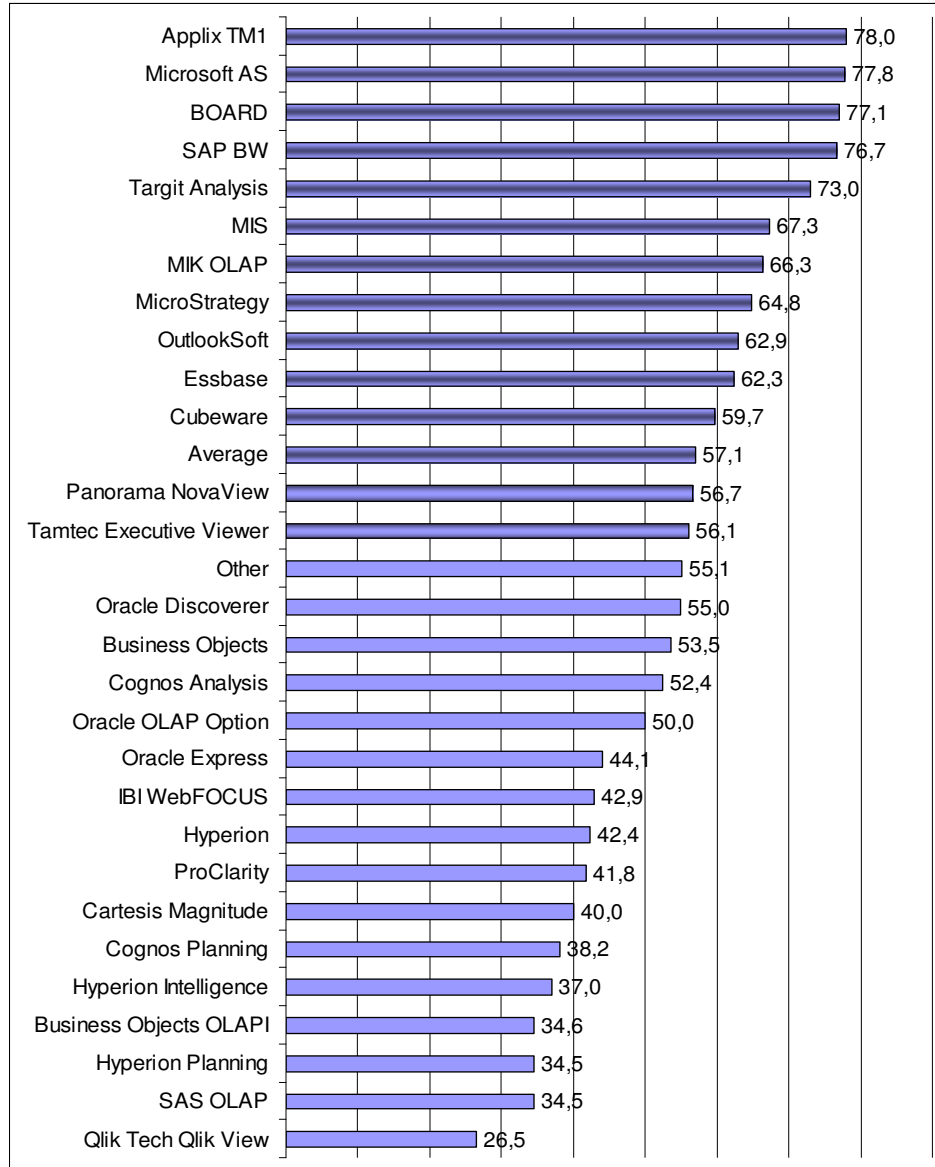
Farklı değerlendirme faktörlerine göre yazılımların fiyatlarının medyan değerleri Şekil 3.6'da görülmektedir[Pendse, 2006]. Kullanım amacı, kullanıcı sayısı, uygulanan teknolojinin türüne göre fiyatlar değişkenlik göstermektedir.



Şekil 3. 6. Ortalama lisans ücretleri (\$)

OLAP Yazılımları -Müşteri Tercih Puanlaması

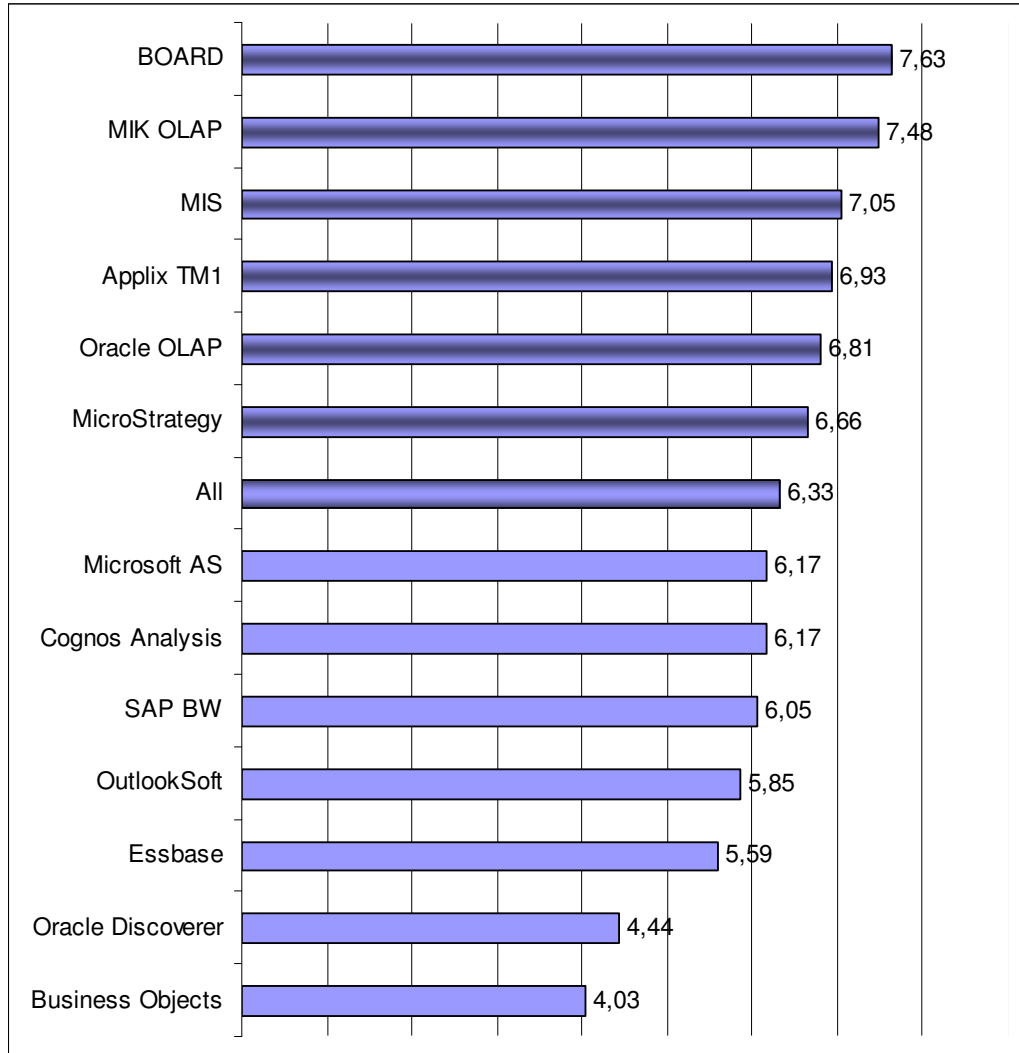
Bir OLAP ürününün başka bir OLAP ürününe tercih edilmesi çok sayıda faktöre bağlı olduğunda karmaşık bir seçim sürecidir. Fiyat, fonksiyonellik, performans, kullanıcı desteği, kullanım kolaylığı ve en önemlisi satış elemanının sahip olduğu satış kabiliyeti düzeyi OLAP yazılımlarının seçimini etkileyen belli başlı kriterlerdir. Şekil 3.7’de işletmelerin ürünü kullandıktan sonra o ürünü tekrar tercih etme oranları gösterilmektedir.



Şekil 3. 7. Değerlendirme sonrası ürün tercih oranları[Pendse, 2006]

OLAP Yazılımları –Ürün Desteği

Yazılım aktif olarak kullanılmaya başlandıktan sonra ön plana çıkan husus kullanıcı verilen destek hizmetlerdir. Şekil 3.8 kullanıcıların destek hizmetleri bakımından yazılımlara verdiği ortalama skorları içermektedir[Pendse, 2006].



Şekil 3. 8. Ürüne göre destek kalitesi oranları

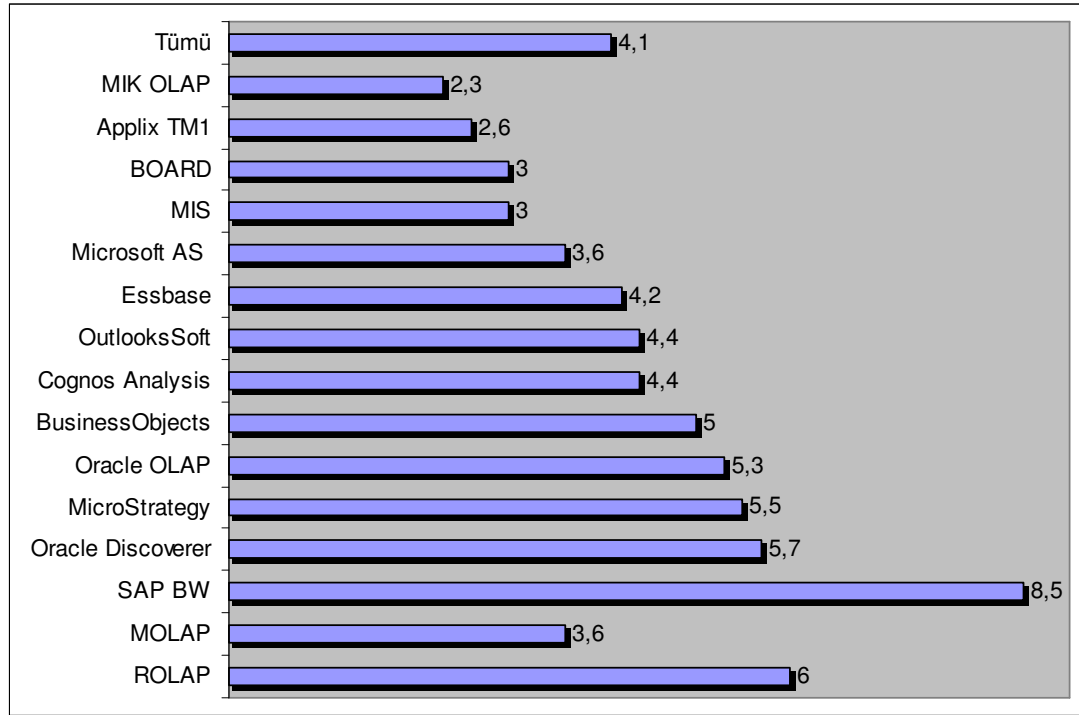
OLAP Yazılımları- İmplementasyon

Yazılımın uygulama geçiş sürecinde gerekli iş gücü maliyetinin yanı sıra, projenin adaptasyonu ve yönetimi sürecinde gerekli işgücü sayısı da önemli bir göstergedir(Şekil 3.9.).

	Ortalama	Ortanca
BOARD	0.946	0.5
MIK OLAP	1.064	0.8
Applix TM1	1.370	1.0
MIS	1.417	1.0
OutlookSoft	2.016	1.5
Oracle Discoverer	2.100	1.0
Microsoft AS	2.229	1.0
Essbase	3.065	1.5
Cognos Analysis	3.197	1.0
MicroStrategy	3.322	2.0
BusinessObjects	4.395	2.0
SAP BW	7.209	3.0
Oracle OLAP Option	7.910	1.5
Tümü	2.719	1.0

Şekil 3. 9. Yazılımın uygulamaya geçirilmesi için gerekli iş gücü [Pendse, 2006]

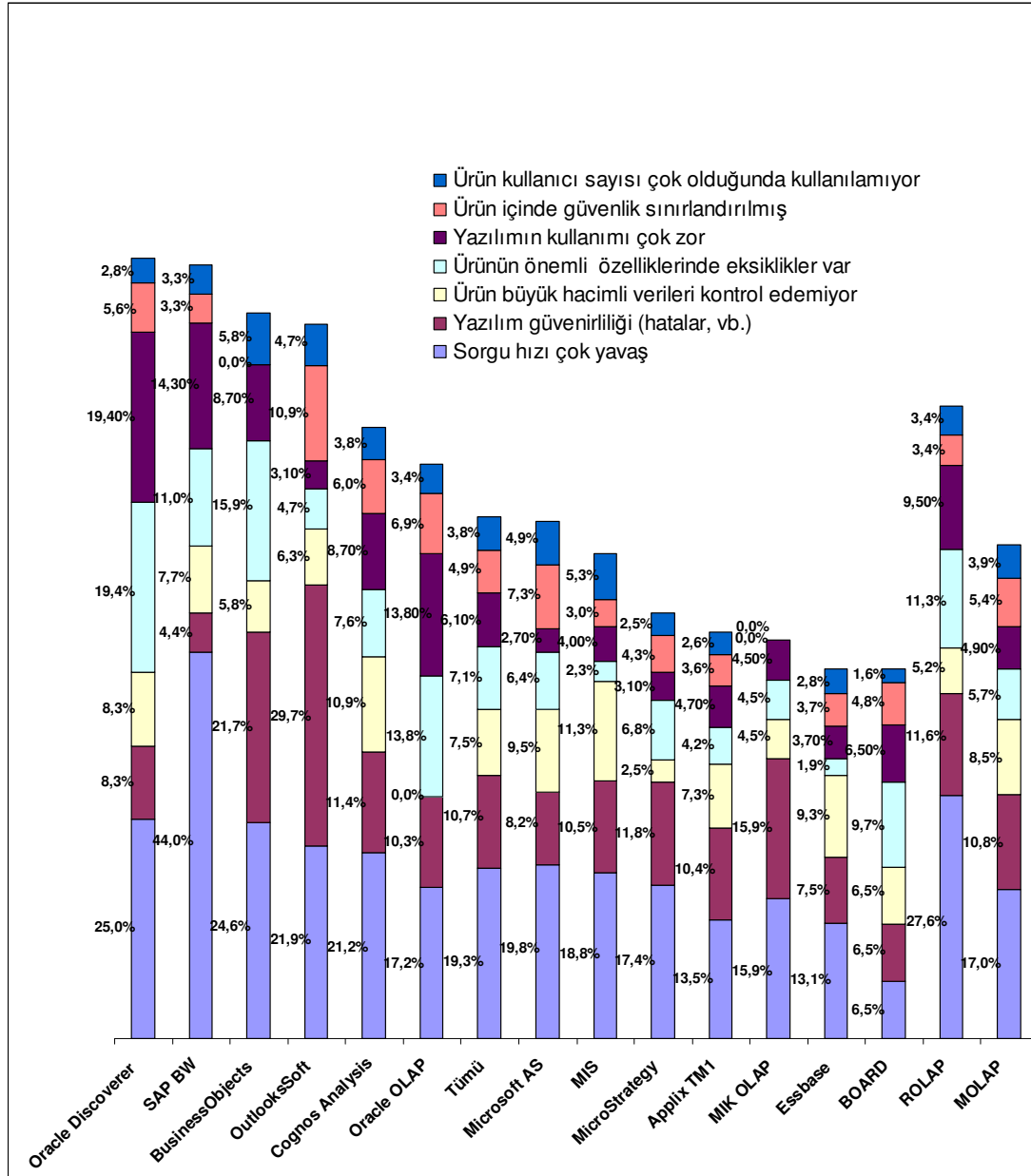
OLAP yazılımlarının uygulama ve adaptasyon süreçlerinin yavaş ilerlemesi en büyük problemlerin başında yer almaktadır. Araştırmalar, uygulama sürecinin 3 ay aşmasıyla birlikte proje başarı şansının da yavaş yavaş düşmeye başladığını belirtmektedir. 13 yazılıma ilişkin ortalama yürütme süreleri ay bazında Şekil 3.10'da sunulmuştur[Pendse, 2006].



Şekil 3. 10. Uygulamaya geçiş süresi (ay bazında)

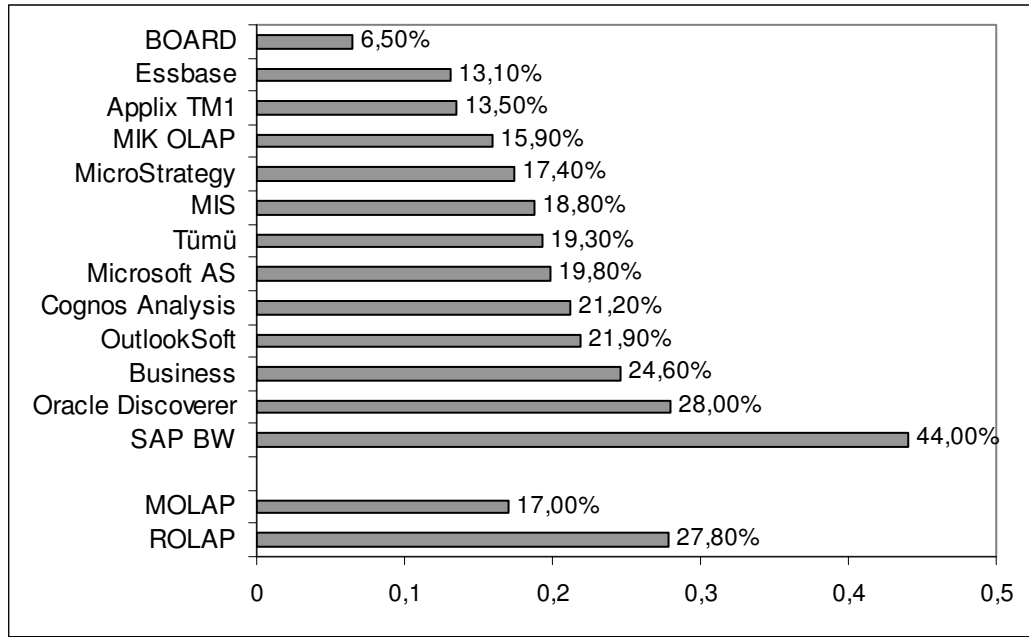
OLAP Yazılımları – Karşılaşılan Problemler

ROLAP ile karşılaştırıldığında, MOLAP ürünlerinde daha az teknik problemle karşılaşıldığı belirtilmiştir. Karşılaşılan teknik problemler sıralandığında tahmin edileceği üzere sorgu performansı hızının düşüklüğü en önemli şikayet olarak belirtilmiştir (MOLAP'ta %17, ROLAP'ta %27,6)(Şekil 3.11).



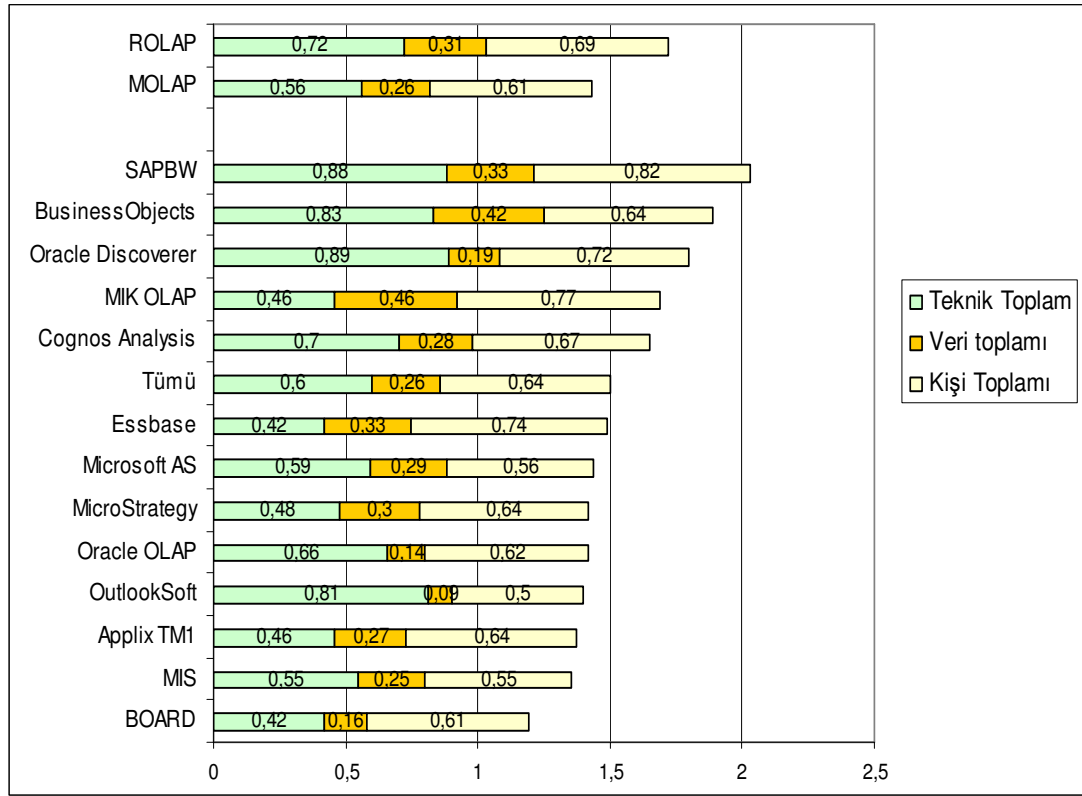
Şekil 3. 11. Raporlanan ciddi problemler [Pendse, 2006]

Sorgu performansı direkt ölçümün yanı sıra kullanıcı beklentilerinin sübjektif olarak değerlendirilmesiyle de incelenmiştir. Şekil 3.12’de sunulan grafik sorgu performans hızı düşüklüğünü problem olarak gören kullanıcıların yüzdelarını göstermektedir.



Şekil 3. 12. Grafik sorgu performans şikayetleri

Teknik problemlerin yanı sıra, insan ve veriden kaynaklanan problemlerde mevcuttur. Bu 3 problem kaynağına göre elde edilen değerlendirme sonuçları Şekil 3.13'de sunulmuştur. Ürünlere göre problem tipi karşılaştırması Şekil 3.13 incelendiğinde görüleceği üzere değişkenlik en fazla veri kaynaklı problemlerden kaynaklanmaktadır (En iyi ile en kötü değer arasındaki oran 4). Teknik ve insan kaynaklı problemler için bu oranlar ise sırasıyla 2,12 ve 1,65'dir. Genel mutlak değerlendirmede ise değişkenlik en fazla teknik problem kaynaklıdır.



Şekil 3. 13. Ürünlere göre problem tipi karşılaştırması

3.6.2. OLAP'ın özellikleri

OLAP zaman kazancının dışında, 3 çok önemli özelliği de beraberinde getirmektedir.

Verilere Çok Boyutlu Bakabilme Özelliği

Analizler sırasında kullanmış olduğumuz her türlü kırıma, boyut adını verebiliriz. Demografik veriler (yaş, cinsiyet, eğitim durumu), sayısal veriler, adetler, işlem miktarları, gerçekleşen ve bütçelenen değerler, ürün tüpleri, ürün özellikleri ve zaman örnek olarak verilebilir. Yöneticiler ve analistler, çalışmalarını sırasında, tüm bu tanımlanan verileri yatay veya düşey eksenlerde karşılaştırarak görmek isteyebilirler.

İlişkisel veri tabanları, bu şekilde raporlara izin vermezler, ancak raporlama araçlarının yetenekleri ile belirli bir noktaya kadar tolere edilebilir. Daha karmaşık

analizler için içine girdiğinde, bir OLAP yapısı kurmadan bu raporları almak imkansız hale gelebilir.

İlişkisel veri tabanları üzerinde karmaşık SQL kodları yazmak, ya da raporlama aracının sahip olduğu programlama dili üzerinde uğraşmak gerekebilir. Bu da, analizi yapan kişilerin, işin özünden çıkarak, analiz gerektirebilecek verilere değil, teknik olanaklara, daha kolay şekilde alabilecekleri verilere kanalize olmaları sonucunu doğurur. Bu nedenle, iş zekası programlarının pratik olmasının yanında, fazla teknik bilgi kullanmadan raporların alınabilir olması, farklı kaynakları bir arada kullanabilecek, konsolide edebilecek yapıda olmaları gerekir.

Boyutların başka bir özelliği de hiyerarşiler tanımlanabilmesidir. Hiyerarşiler sayesinde, hem toplamlara ulaşmak kolaylaşmakta, hem de farklı gruplar için, farklı senaryolar hazırlayabilme şansı doğmaktadır.

Karmaşık Hesaplamalar

Bir OLAP sisteminin gerçek performansı, karmaşık hesaplamaları yapma gücü ile ölçülebilir. OLAP sistemleri, sadece toplama işleminden başka işlemler de yapabilecek güçte olmalıdırlar. Gerçek hayat, her zaman daha karışıktır. Analiz yapanlar için, yüzdesel dağılımlar, rakamlar çok daha önemlidir. Birkaç yıllık satış içerisinde, binlerce ürün türü için günlük bazda satışları yüzdesel olarak analiz edip, sıraya dizebilmek, bir İlişkisel Veritabanı Yönetim Sistemi(İVTYS) ile saatler sürecektir bir raporun çalışmasını gerektirebilir. Oysa uygun bir OLAP sistem ile, bir günlük satışlar ve birkaç yıllık satış rakamı arasında bir fark olmamalıdır. Satış tahminlerinde, genellikle “hareketli ortalama” ve “yüzde artış” gibi trend analizleri kullanılır. Finansal analizlerde, envanter hesaplarında ve portföy performans hesaplarında, zamana göre ürünlerin toplanma sırası, sonucu tamamen değiştirebilir (yukarıdan aşağıya, ya da aşağıdan yukarıya, LIFO-FIFO). Kullanılacak OLAP yapısında, bu şekilde hesaplamalara da izin verir bir yapısının olması gerekir.

Zaman Kavramları

Zaman boyutu, neredeyse her analizin temel bileşenidir. Zaman, diğer boyutlardan farklı olarak, kendine has bir sıralama içerisinde gider. Alfabetik (Ocak her zaman Şubat'tan önce gelmelidir) veya nümerik sıralamalardan (12/31, 01/01'den önce gelmelidir) her zaman farklıdır. Gerçek OLAP sistemleri, zamanın bu şekilde sıralanmasını sağlarlar.

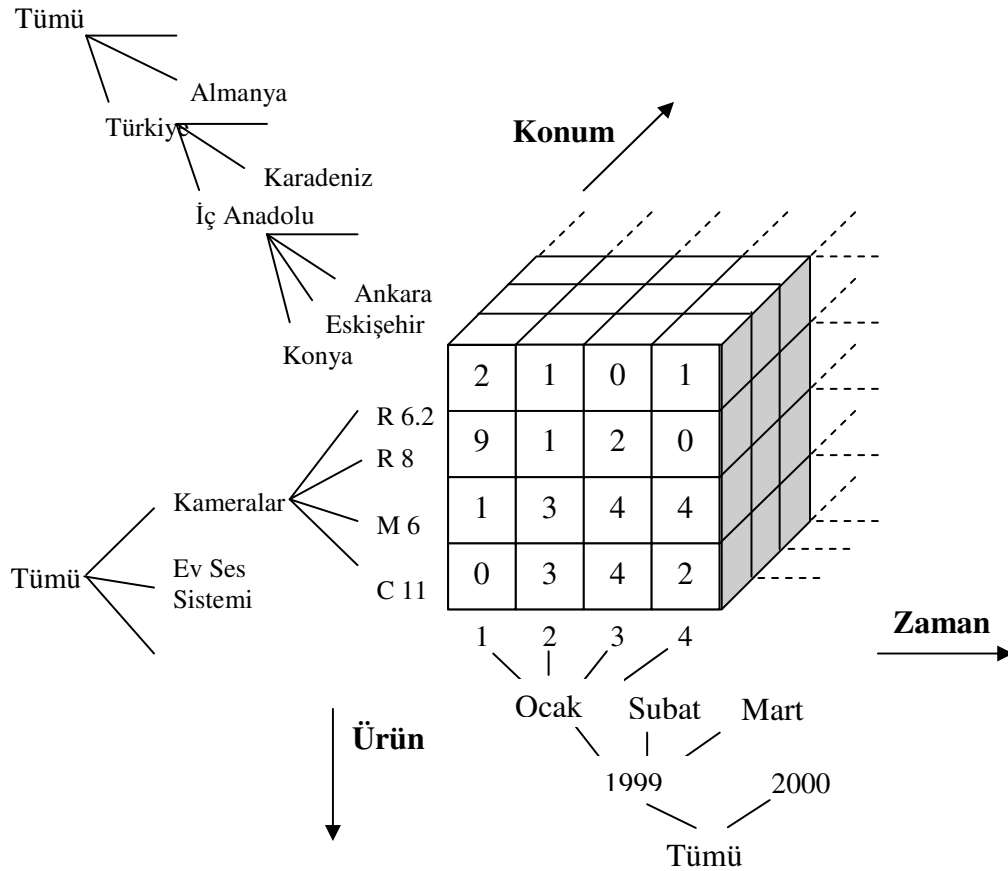
OLAP ın üstünlükleri, daha önce bahsedilen özelliklerinden gelmektedir. OLTP ve OLAP mukayesesi Çizelge 3.2' de gösterilmiştir.

Çizelge 3. 2. OLTP ve OLAP karşılaştırması [Han ve Kamber, 2001]

KRİTER	OLTP	OLAP
Karakteristik	Operasyonel İşleme	Bilgi İşleme
Oryantasyon	İşlem	Analiz
Kullanıcı	Satıcı, VT yöneticisi, VT uzman	Bilgi Çıkarımcısı (uzman vb.)
Fonksiyon	Günlük Operasyonlar	Uzun dönemli bilgi gereksinimi, Karar Desteği
Veri Tabanı Tasarımı	ER tabanlı, Uygulamaya yönelik	Yıldız/Kartanesi, Özneye yönelik
Veri	Günlük ve izole	Kronolojik, birleştirilmiş
Özetleme	İlkel, Yüksek derecede detaylı	Özetlenmiş, Birleşik
Şema	Detaylı, Düz ilişkisel	Özetlenmiş, Çok boyutlu
Birim İş	Kısa, basit işlemler	Kompleks sorgular
Erişim	Oku/Yaz	Genellikle oku
Odak	Veri girişi	Bilgi çıkışı
Kullanım	Yapısal, Tekrarlı	Önceden tanımlanmış
Operasyonlar	Birincil anahtarla indeksleme	Bir çok gözden geçirme
Erişilen kayıt sayısı	Onlar düzeyinde	Milyonlar düzeyinde
Kullanıcı sayısı	Binlerce	Yüzlerce
Veri Tabanı Büyüklüğü	100 MB => GB	100 GB => TB
Öncelik	Yüksek performans, mevcudiyet	Esneklik, son kullanıcı bağımsızlığı
Metrik	İşlem (Transaction) verimi	Sorgulama verimi, Cevaplama süresi

3.6.3. Veri Küpü (VK)

“Veri Küpü” terimi veri ambarcılığı literatüründe sürekli olarak kullanılmaktadır. Bir veri kümesi; çok boyutlu hiper-karmaşık kavramsal modellemedir, ya da kısaca *veri küpü*’dür. Kümedeki bir veri birimini tarifleyen d fonksiyonel özellikleri boyutlardır. d boyutu “dimension” göstermektedir. Bunların bazıları hiyerarşik (örneğin; zaman boyutunda yıl- çeyrek yıl- ay- gün gibi), bazıları var olan bir özellik için çoklu hiyerarşiye (Örneğin yine zaman boyutunda hafta,-gün gibi) sahiptir. Bir formdaki kayıtları oluşturan veri kümesi (d -tuple) değerleri çok boyutlu dizilerdeki bir hücreyi temsil etmektedirler. Hücreler, ekstra özelliklerdeki değerleri içerirler.



Şekil 3. 14. Veri küpü örneği [Han ve Kamber, 2001]

Basit bir örnek; Ürün, Zaman (Satış zamanları) ve Lokasyon (Satış yerleri) boyutları ve de ölçülebilir özelliklerin (satış işlemleri değerlerinin) yer aldığı bir satış küpüdür. Böylece her bir ürün ile zaman ve konum için satışların uygun miktarları elde tutulmaktadır. Şekil 3.14 tüm özellikleriyle boyut hiyerarşisini gösteren bir örnektir. Satış işleminin olmadığı bir hücre varsa bu hücre boştur. Şekilde boş hücreler “0” değerini almıştır.

Veri küpleri üzerindeki yaygın, kabul gören sorgu tipleri genelleme (roll-up) ve derinleme (drill-down) (artış, sırasıyla azalış, bir ya da daha fazla özelliğin hiyerarşi boyunca bütünleşme seviyesi) ve dilimleme- küp şeklinde kesim (slice-and-dice) (özelliklerin ve ilgili hücrelerin seçimi), Chaudhuri ve Dayal (1997) tarafından özetlenmiştir.

Orijinal küp terimi, kübün boyutlarıyla belirlenen d-boyutlu uzayda tariflenen bir veri kümesinin açık izdüşümü gibi anlam ifade etmektedir. Eğer küpün herhangi bir hücresi hesaplama sonucunda ya da bütünleştirmeyeyle ortaya çıkmış bir değer içeriyorsa (örneğin; ara toplamalar), işlenmiş küp (preaggregated cube) olarak isimlendirilmektedir.

Küp(Cube) 1997’de Gray ve arkadaşları tarafından Yapısal Sorgulama Dili (Structured Query Language (SQL))’nin gruptama (GroupBy) işlemcisinin geliştirilmesiyle ortaya çıkarılmıştır. Küp belirlenmiş boyutların tüm alt kümelerindeki veri kümelerinin gruptama sonuçlarını hesaplayabilmektedir. Böylece 2^d kadar grup oluşmaktadır. Her bir grup küboid (cuboid) ve tüm sonuçlar ise küp olarak kullanılmaktadır. Şekil 3.15’ de birleştirme operatörü Topla (SUM)’nın boyutlarda uygulanması örneği gösterilmektedir. Zaman (Z), Ürün (U), Lokasyon (L) ve ölçülebilir değerlerin miktarı (A) olarak tariflenmiştir. İlişkisel veri tabanı terminolojisinde küboidler GROUP BY sorgularına benzer olarak oluşarak görünmektedir. (Şekil 3.16)

Kolaylıkla küboidler söz konusu olduğunda boyutların harflerini kullanabiliriz. Örneğin, küboidten söz edildiğinde <ZU> harfleri Zaman ve Urun için gruptama

sonuçlarıdır. Benzer şekilde <ZUL> en detaylı bilgiyi içermektedir ve temel küboid (base cuboid) olarak ta söylenmektedir.

Küpün esas amacı bütünleşik sorguları desteklemektir. Bir sorgu sadece bir kaç özelliği seçerek ya da kendine ayrılan küboidi işleyerek verimli cevaplar çıkarabilir. Örneğin 1999 yılının toplam satışlarını bulmak için <Z> küboidine göz atma maliyeti olacaktır, aynı sonuçları hesaplaması için temel küboidte arama, bulma ve çoklu veri kümelerinin birleştirilmesi gerekmektedir.

Z	U	L	A
99	p1	SB	2
99	p1	SF	3
99	p2	LA	1
00	p1	SB	2
00	p1	SF	3
00	p2	LA	4
01	p1	SF	3
01	p2	LA	1

GROUP BY Z.U.L.

Z	U	A
99	p1	5
99	p2	1
00	p1	5
00	p2	4
01	p1	3
01	p2	1

GROUP BY Z.U.

Z	L	A
99	SB	2
99	SF	3
99	LA	1
00	SB	2
00	SF	3
00	LA	4
01	SF	3
01	LA	1

GROUP BY Z.L.

U	L	A
p1	SB	4
p1	SF	9
p2	LA	6

GROUP BY U.L.

Z	A
99	6
00	9
01	4

GROUP BY Z.

U	A
p1	13
p2	6

GROUP BY U.

L	A
SB	4
SF	9
LA	6

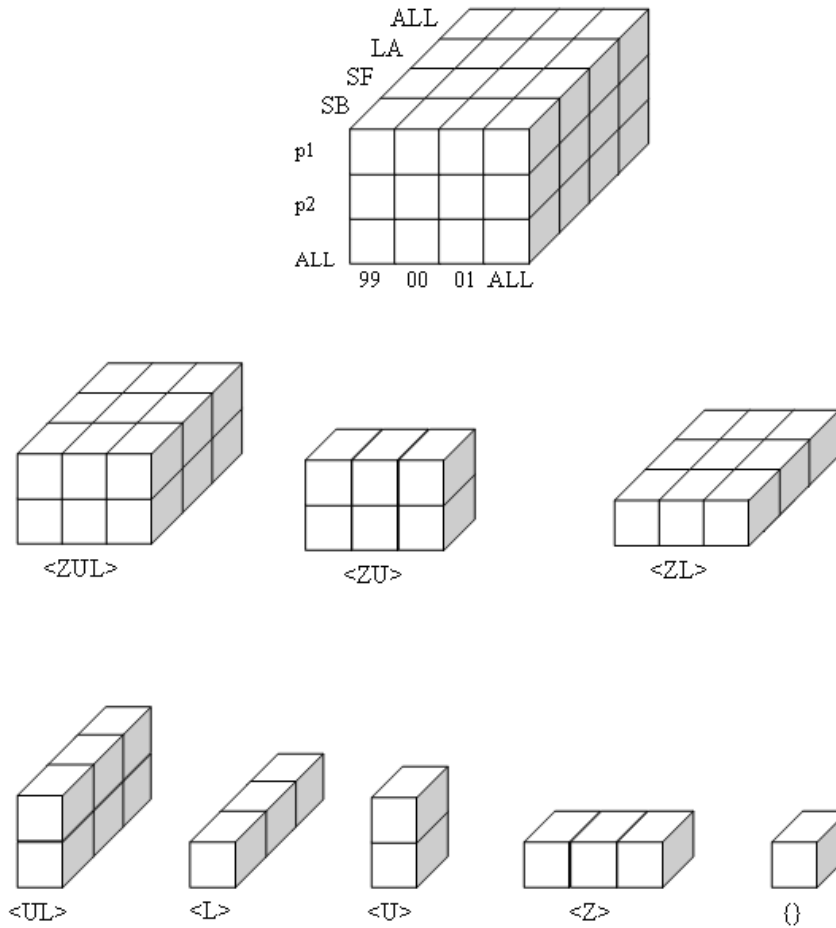
GROUP BY L.

Şekil 3. 15. Bir üç boyutlu örnek küpün küboidleri

Küpte uzmanlaşmış dizin yapılarında, Johnson&Shasha(1997,1999) ve Roussopoulos ve ark. (1997), sorgulama zamanlarını daha da azaltmışlardır.

Küp ortaya çıkarıldıktan sonraki araştırmaların çoğu, küpün üzerinde verimli hesaplamalar ve kullanılan alanın azaltılması yönünde olmuştur. Bayer ve

Ramakrishnan (1999), küboid oluşturma ya da parçalanması alanında bazı kısıtlamalar yapan ve bütünleştirmeyi yeterli seviyede kesen bir teknik önermişlerdir. Diğer araştırmacılar, küp hesaplamalarını bir optimizasyon problemi gibi formüle etmişlerdir (Örneğin, Gupta ve arkadaşları; Harinarayan, Rajaraman, & Ullman, 1996 ;Shukla, Deshpande, & Naughton, 2000; Smith ve arkadaşları., 1998). Bu çalışmaların amacı, depolama kısıtıyla birlikte sorgulama maliyetini minimum tutarak küboidlerin alt kümelerinin seçilmesi ve gerçekleştirilmesidir.

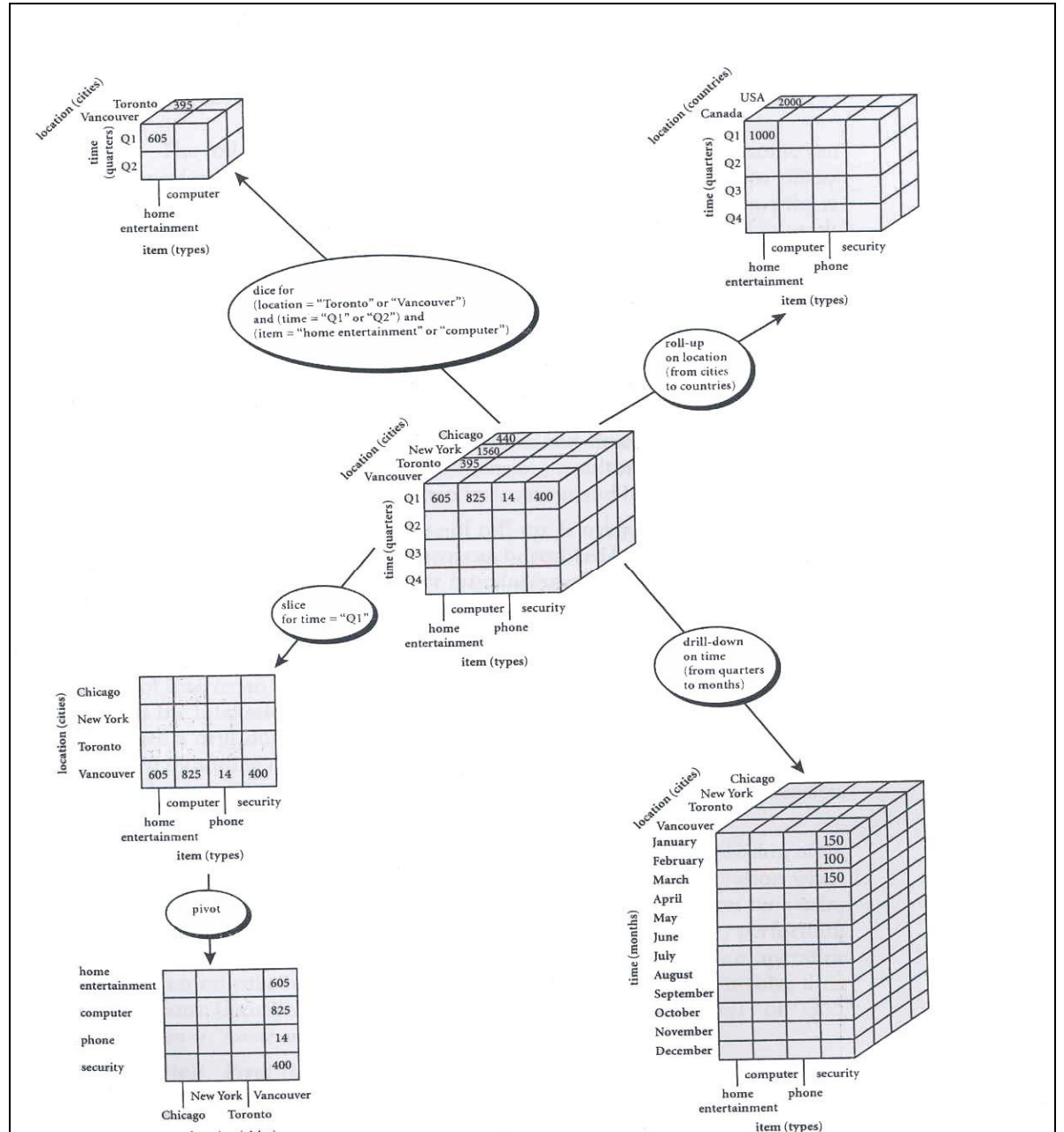


Şekil 3. 16. Şekil 3.15'e bağlı olarak Küboitleri gösteren bir çokboyutlu VK

Daha sonraki çalışmalarda küpün bakımının önemi işaret edilmektedir. Sonuç olarak, Baralis ve ark. (1997), Gupta (1997), ve Gupta ve Mumick(1999) ilk çalışmalarında optimizasyon problemlerinin güncelleme maliyetleri üzerine odaklanmışlardır.

3.6.4. Çok boyutlu veri üzerinde OLAP işlemleri

Şekil 3.17’de şehirlere göre; ev eğlence sistemleri, bilgisayar, telefon ve güvenlik bilgilerinin bulunduğu çok boyutlu bir veri modelinden açıklayıcı parçalar örnek olarak seçilmiştir.



Şekil 3.17. Çok boyutlu veri üzerinde tipik OLAP işlemleri örneği [Han ve Kamber, 2001]

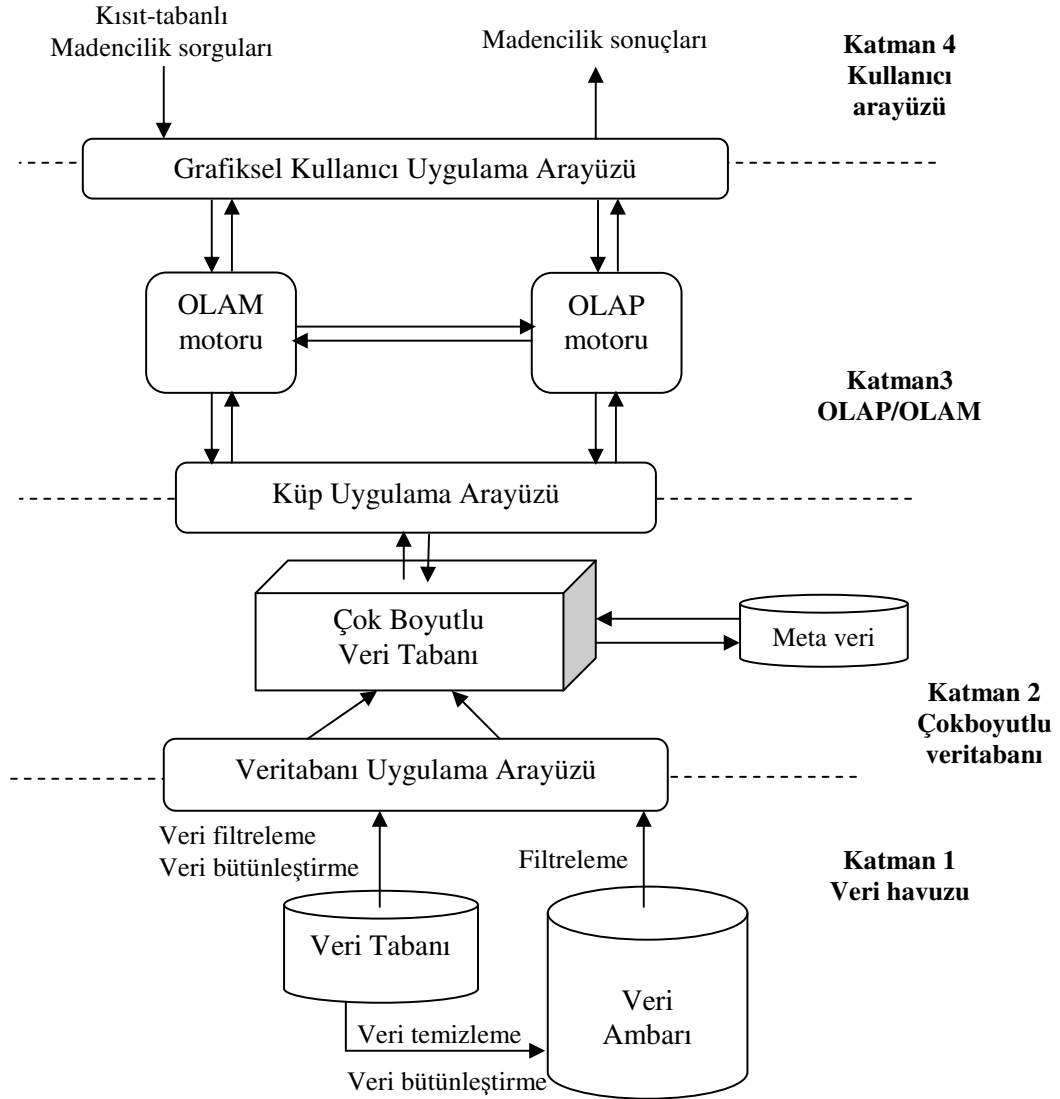
Şekil 3.14’de verilen OLAP işlemleri örneğinde; aşağıda tariflenen her bir OLAP işlemi gösterilmiştir. Şeklin merkezinde elektronik ürün satışlarını gösteren 3 boyutlu bir VK yer almaktadır. VK’nün boyutları konum (location), zaman (time) ve ürünlerden (item) oluşmaktadır. Konum boyutu şehir bazında, zaman boyutu çeyrek yıl (Q1,Q2,Q3,Q4) ve ürün boyutu ise tip bazında gruplandırılmıştır. Merkezde yer alan VK’nde Q1 ilk çeyrek yılı, konum Chicago, New York, Toronto ve ürünler home ent., computer, phone, security referans olarak alınmıştır. Hücrelerde yer alan hesaplama değeri ise dolar olarak satış rakamlarıdır.

Genelleme (Roll-up): Genelleme işlemi VK üzerinde, hiyerarşide üst seviyeye çıkarak ya da boyut azaltarak toplama işini gerçekleştirir. Verilen örnekte, genelleme işlemi konum boyutunda gerçekleştirilmiştir. Şehirler gruplandırılarak, bir üst seviyede yer alan ülke seviyesine çıkılmıştır. Şehirlerde gerçekleşmiş satışlar toplanarak örnekte görüldüğü gibi ilk çeyrek yılın USA satış rakamları elde edilmiştir.

Derinleme (Drill-down): Derinleme genellemenin tersidir. Verinin daha fazla detaylandırılması amacıyla kullanılır. Hiyerarşide bir alt seviyeye inme ya da boyut artırma işlemini gerçekleştirir. Örnekte derinleme işlemi zaman boyutunda gerçekleştirilmiştir. Satış rakamları çeyrek yıldan, aylık seviyeye getirilmiştir. Daha özet bilgiye sahip olan çeyrek yıldan, daha detaylı aylık verilere ulaşılmıştır.

Dilimleme ve Küp(Zar) çıkarma (Slice and Dice): Dilimleme operasyonu, mevcut küpte işlem için küpün bir boyutunun seçimi için kullanılır. Şekil 3.14’de yer alan örnekte dilimleme operasyonu referans küp üzerinden diğer boyutlar aynı kalırken, zaman boyutunda sadece Q1 için satış rakamlarının elde edildiği durumu göstermektedir. Küp çıkarma işlemi ise iki ya da daha fazla boyut seçilerek alt küpün elde edilmesi için kullanılmaktadır. Verilen örnekte; (konum=“Toronto” veya “Vancouver”) ve (zaman=“Q1” veya “Q2”) ve (ürün= “home ent.” veya “computer”) kriterleri için alt küp oluşturulmuştur.

Eksende çevirme (Pivot): Pivot, verinin alternatif gösterimi ve farklı eksenlerden görüntüsü için kullanılan bir görselleştirme işlemidir. Örnekte; 1. çeyrek yıl (Q1) sabit eksen olarak belirlenmiş, konum ve ürün için dilimleme işlemi ile verinin gösterimi yapılmıştır. Sonrasında ise pivot işlemi ile boyutların yerleri değiştirilerek, iki boyutlu ürün-konum tablosu elde edilmiştir.



Şekil 3. 18. Bütünleşik bir OLAM(Çevrimiçi Analitik Madencilik) ve OLAP yapısı [Han ve Kamber, 2001]

Veri madenciliği genel bir kabulle insan merkezli bir süreçtir. Örüntü ve bilginin otomatik olarak keşfindeki veri analizi sürecinde, VM sistemine sorulmasından

ziyade kullanıcı sık sık sistemle etkileşim halindedir. OLAP etkileşimli veri analizi ve keşifçi VM için gerekli hazırlıkların sağlanmasında iyi bir örnektir oluşturabilir. OLAP işlemleri kullanılarak, ürün ve kullanıcılar arasındaki ilişkinin keşfi yapılabilir. Belli bir ürün tipi için genelleme işlemi ile o ürüne ilgisi olan belirli müşteri özellikleri ortaya çıkarılabilir. OLAM sunucusu veri küplerinde online işlemde OLAP sunucusunun online işleme mantığıyla aynı performansı gösterir. Şekil 3.18 de bu amacı gerçekleştirmek için kullanılacak bütünleşik yapı gösterilmektedir.

3.6.5. OLAP Sunucu Tipleri

MOLAP (Çok Boyutlu OLAP)

Veri gruplamaları ve toplamalarını tüm boyutlarıyla hesaplayarak tuttuğu yapıdır. Tüm hesaplamalar yapıldığı için diğer yapılardan daha hızlı işler.

ROLAP (İlişkisel OLAP)

Bu OLAP tipinde veri ilişkisel veritabanları üzerinde kalır. Dinamik çokboyutlu verilerin analizinde tercih edilir. Gerçek zaman (Real-time) veri güncellemelerini destekleyebilir. Ancak performansı daha düşük olabilir. Çok kullanıcıli sistemlerle yapılan analizler özellikle çok fazla yere ve güçlü makinelere ihtiyaç duyabilir.

HOLAP (Hibrit OLAP)

Kısaca MOLAP ve ROLAP 'ın karışımı olarak tanımlanabilir.

4. VERİ KÜPLERİNİN BÜTÜNLEŞİK KULLANIMINA YÖNELİK YENİ BİR OLAP MİMARİSİ

1997’de Gray ve arkadaşları tarafından SQL’in gruptama (GroupBy) operatörünün genel kullanımı üzerine yapılan bir çalışmayla veri küpü literatüre girmiştir. 3 boyutla kullanılan yapılar çok boyutlu küp (multicube), 3 boyutun üzerinde bir yapıda olan küpler hiperküp (hypercube) olarak adlandırılmaktadır.

OLAP küplerinin kullanılmasına yönelik ilk çalışmadan günümüze kadar geçen zamanda, uygulama mantığı değişmemiş fakat her alanda olduğu gibi bu kısımda da kullanım kolaylığı ve bilgi çıkarımını iyileştirecek çalışmalar yapılmıştır. Arayüzler olmasına rağmen arka planda veritabanı işlemlerini destekleyen kodlama dili ya da sorgulama dili kullanılmaktadır.

4.1. Literatur Taraması

Literatürde OLAP kapsamında VK ve VM ile ilgili çalışmalar aşağıda verilmiştir.

OLAP teriminin ilk olarak ortaya çıkışı 1993 yılında, Dr. E.F.Codd ve ark. ortaya koyduğu kurallar çerçevesinde olmuştur. Bu kurallar, OLAP için bir temel oluştursa da, kimi çevrelere göre, o yıllarda Arbor Software (Şimdiki Hyperion Solutions) için bir white paper olmaktan öteye gidememiştir. Fakat günümüzde çoğu çalışmaya referans olduğu görülmektedir. Çalışmada OLAP terimi, sistem bileşenleri ve özellikleri açıklanmıştır. OLAP’ın geleneksel veri tabanlarına göre, stratejik işletme kararlarında ki üstünlükleri, önemi ve zaman uygunluğu üzerinde durulmuştur [Codd ve ark., 1993].

Gray ve ark. (1997) tarafından yapılan çalışma VK ya da basitçe küp operatörünün önerildiği ilk çalışmadır. Yapısal Sorgulama Dili (Structured Query Language-SQL)’nin gruptama (Group By) işlemcisinin genelleştirilmesiyle ortaya çıkarılmıştır. Dökümanda roll-up operatörü ve SQL’e uygulanışı, kullanıcıların küpler için yeni

bütünleştirme fonksiyonları nasıl tanımlayacakları açıklanmış ve küp hesaplamasındaki verimli teknikler tartışılmıştır [Gray ve ark., 1997].

Çevrimiçi Analitik Madencilik (Online Analytical Mining-OLAM); OLAP teknolojisi ile veri madenciliği tekniklerini farklı büyüklüklerdeki veri tabanları, veri ambarlarını kullanarak farklı seviyelerdeki özet raporları ve sonuçları çıkaran bir mekanizmadır. Veri tabanı endüstrisinde OLAP teknolojisi ve veri ambarcılığında ki hızlı gelişmeler, OLAP madenciliğindeki mekanizmanın gelişimine umut vermektedir. Han (1997) tarafından yapılan bu çalışmada OLAP madenciliği bileşenleri ilk kez ortaya konmuştur ve OLAP'ın veri madenciliği sistemine nasıl uyarlanacağını tartışmıştır.

Çalışmada Gupta ve ark. (1997) tarafından özet tabloların ve indekslerin seçimini otomatize eden algoritmalar verilmiştir. Temelde, zaman karmaşıklığını artıran bir algoritma ailesi hazırlanmış ve güçlü performans sınırları ortaya konmuştur. Daha yüksek karmaşıklıkta algoritmalar ile daha iyi performans sınırları elde edilmiştir. Bununla birlikte, performans sınırındaki artış azalmıştır ve orta karışıklıkta kullanılan bir algoritmanın optimale yakın sonuç verdiği gösterilmiştir [Gupta ve ark., 1997].

Veri küpünde ön hesaplamalı yapılan toplam işlemleri OLAP uygulamalarında etkin sorgu işleme sağlar. Goil ve Choudhary (1997), dağıtık hafıza paralel bilgisayarlarda veri küplerinin inşa edilmesine yönelik bir algoritma sunmuştur. Veri, bir ilişkisel veri tabanından çokboyutlu dizi yapısına aktarılmaktadır. Performans karşılaştırması yapmak üzere sıralı tip (sort-based) ve karışık tip (hash-based) olarak iki yöntemle temel küp oluşturulmuştur. Boyut hiyerarşisinde roll-up operasyonu ve küplerin VM'nde kullanımına yönelik özellik odaklı bir örnek uygulama yapılmıştır [Goil ve Choudhary, 1997].

Veri ambarlama ve OLAP teknolojisi ile ilgili ihtiyaçları vurgulayan bir gözden geçirme çalışması Chaudhuri ve Dayal (1997) tarafından yapılmıştır. Özetleme, temizleme, ve bir veri ambarına veri yükleme, çokboyutlu veri modeli OLAP, sorgu

ve veri analizi için kullanıcı araçları, etkin sorgu işleme için sunucu uzantıları ve veri ambarı, metadata, yönetici için araçlar tanımlanarak incelenmiştir [Chaudhuri ve Dayal, 1997].

Vassiliadis (1998) çok boyutlu veri tabanı için bir model önermiştir. Boyutlar, boyut hiyerarşileri ve küplerin modeldeki yapısı biçimsel olarak anlatılmıştır. Yaklaşım, temel küp kavramından yola çıkarak, karmaşık operasyonlarla küp operasyonları sonuçlarının hesaplanmasını ele almıştır. Dikkat edilmesi gereken nokta, küpün seri işlemlere imkan sağlamasıdır [Vassiliadis, 1998].

Müşteri davranışlarının profilinin çıkarılması; dolandırıcılık tesbiti, hedef pazarlama promosyon vb. çoğu uygulamada artan bir öneme sahiptir. Müşteri davranış profilleri çok büyük hacimli işlenebilir veri kümelerinden ortaya çıkmaktadır. Bu durum, veri ambarı ve OLAP tabanlı, ölçeklenebilir, esnek tabanlı bir profil çıkarma motoru geliştirmeyi gerektirmiştir. Chen ve ark. (1999) yaptıkları çalışmada; müşteri profilleri olasılık dağılımını çok boyutlu, çok seviyeli veri küpleri üzerinde OLAP operasyonlarını kullanarak belirlemiştir. Edinilen tecrübe, profil çıkarımı ve desen analizinde OLAP temelli çözümlerin basitliği ve güçlü olduğudur [Chen ve ark., 1999].

Datta ve Thomas (1999), basit fakat geliştirilebilir bir veri küpü modeli önermişlerdir. Küp üzerinde OLAP operasyonlarını destekleyici nitelikte basit bir matematiksel yapı kurulmaya çalışılmıştır [Datta ve Thomas, 1999].

Theodoratos ve Sellis (2000), çok boyutlu veri tabanları için basit bir veri modeli ve temel OLAP operasyonlarının modellenmesine izin veren bir basit cebirsel çok boyutlu sorgulama dili önermiştir [Theodoratos ve Sellis, 2000].

Lin ve Kuo (2000) OLAP veri küplerinin seçiminin otomatikleştirilmesinde genetik algoritma kullanmıştır. Detayda, kısıtların ele alındığı teknikler üzerinde yoğunlaşmıştır [Lin ve Kuo, 2000].

Palpanas (2000), veri madenciliği teknikleri kullanılarak, veri ambarlarındaki çok çeşitli veri yerleşimlerinden, ilginç bilgilerin çıkarımı ve tanımlanması problemi araştırmıştır. OLAP, VM birlikteliği ve sorgu cevaplama literatürü ele alınmıştır. Geleneksel VM tekniklerinin OLAP sistemlerine adaptasyonundaki gereksinimleri tartışılmıştır [Palpanas, 2000].

Thalhammer ve ark.(2001) “aktif veri ambarı” olarak adlandırılan veri ambarında karar vermeyi otomatikleştirmeye çalışan yeni bir yaklaşım sunmuşlardır. Aktif veri ambarındaki temel fikir, veri analizi ve karar verme için ECA (event/condition/action) kurallarının çok boyutlu özelliklerini kullanarak genişletmektir [Thalhammer ve ark., 2001].

Elektronik ticaretin kapsadığı alanın başarılı VM için tüm içeriğe sahip olduğundan yola çıkarak; Ansari ve ark. (2001) bir yazılım desteğiyle bütünleşik mimari tariflemiştir. Yapının, ön işleme, temizleme ve verilerin anlaşılması için harcanan efordan %80 zaman tasarrufu sağlandığı iddia edilmektedir. Mimari veri tanımlama, müşteri etkileşeni ve analiz katmanlarından oluşmaktadır. Bu katmanlar birbirlerine veri transfer köprüleri ile bağlı bir yapıya sahiptir [Ansari ve ark., 2001].

Lau ve ark. (2001) OLAP küp mimarisini açıklayarak, şekillendirmiş ve sonrasında küp işlemleri kullanılarak elde edilen raporlarla CLIPS uzman sistemi yazılımıyla tedarikçi seçimi yapmışlardır [Lau ve ark., 2001].

Chau ve ark. (2002) tarafından geliştirilen tasarım yönetimi Karar Destek Sistemi KDS (Construction Management Decision Support System-CMDSS) prototipi, veri ambarı teknolojisi ile OLAP’ın birlikte çalıştırılmasını tariflemektedir. Bir veri ambarının ve KDS’nin bir araya getirilmesi doğru veri kullanarak ihtiyaç duyulan bilgiye direk, hızlıca ve mantıklı bir yolla erişimi sağlayacaktır. Tasarım yöneticilerinin farklı açılardan bakabilmeleri için çok boyutlu veri küpleri kullanılarak bir KDS oluşturulmuş, küçük bir örnek üzerinde denenmiştir. [Chau ve ark., 2002].

Alhajj ve Kaya (2003), nicel özellikler içeren bir bulanık veri küpünün bilgi keşfinde kullanımına yönelik bir genel mimari önermişlerdir. Temel yapıdaki küp kullanılarak benzer bulanık bileşenler içeren, çok boyutlu çevrimiçi ilişki çıkarımına yönelik olarak OLAP VM uygulaması yapılmıştır [Alhajj ve Kaya, 2003].

Maniatis ve ark. (2003), OLAP verileri için literatürdeki yapı haricinde yeni bir küp hazırlama yöntemi sunmuşlardır. Öneri, küpün daha önceki mantıksal modellerindeki çok karmaşık durumlara değinmektedir. Mantıksal model ve yeni yöntemle hazırlanan modelin tipik OLAP işlemlerin de kolayca nasıl eşleşeceği gösterilmiştir [Maniatis ve ark., 2003].

Leung ve ark. (2003), geçmiş tecrübelerden elde edilmiş uzman tavsiyeleri ile optimizasyon sürecine destek veren bir uzman sistem önermişlerdir. Bulanık temeli ile Genetik Algoritma ve OLAP'tan ismini alarak ortaya çıkan Uzman Sistem, 3 teknolojiyi içermektedir; i) insan düşünme ve karar mekanizması için bulanık mantık, ii) analiz edilen bilginin optimizasyonu için genetik algoritma ve iii) geleceğe yönelik karar vermede kullanılan bulanık kararları VM süreçleri ile çıkaran mekanizmaya destek veren OLAP. Değerlendirme için, iyon kaplama sürecinin optimizasyonunda bir uygulama yapılmıştır [Leung ve ark., 2003].

Lin ve Kuo (2004), küp seçim probleminde genetik algoritma uygulamışlardır. Genetik-Greedy hibrit yöntemini öneren çalışmada; benzer depolama kısıtları altında, diğer yöntemlere göre, büyük oranda sorgu ve küp bakım maliyetlerinin azalacağı ileri sürülmektedir. [Lin ve Kuo, 2004].

Hu ve Cercone (2004), web kullanımlı madencilik ve zeki iş raporlama için genel bir VA/OLAP çatısı sunmuştur. Çatı bileşenleri, veri yakalama-web evi oluşumu-desen keşfi, küp inşası, desen değerlendirme ve düzenleme olarak 4 kısımdan meydana gelmiştir. Önerilen VA/OLAP çatısı bazı ticari e-alışveriş sistemleri ile birleştirilmiş ve örneklenmiştir [Hu ve Cercone, 2004].

Vaisman ve ark.(2004)MDDLX(Multidimensional Data Definition Language) olarak adlandırılan çok boyutlu veri tanımlama dilini geliştirmişlerdir. Çok boyutlu ifadesi MDX uzantısı olan ve veri küpünde boyut ve hiyerarşisini güncellemeye imkan sağlayan yaklaşım, küçük bir uygulama ile açıklanmıştır [Vaisman ve ark., 2004].

Tseng (2005), yeni bir kapsama yönelik olarak çok boyutlu deyimler(MDX-Multi Dimensional eXpressions) yapısıyla ilgili, SQL benzeri bir dil önermiş ve sayısal veri haricinde, text datalarının küp mimarisinde sorgulanmasına yönelik bir çalışma yapmıştır [Tseng, 2005].

Kaya ve Alhajj (2005), yeni çok etmenli bir yaklaşım öne sürmüştür. Yaklaşım, modüler yardımcı öğrenme sistemlerinde, madencilik süreçlerinin kullanımına yöneliktir. Etmenlerce raporlanan bilginin etkin bir şekilde işlenebilmesi için, bulanıklık ve OLAP temelli madenciliği bir araya getiren bir bakış açısıdır. OLAP mimarisine uygun olarak etkin işleme ve saklamaya imkan veren bir bulanık veri küpü tanımlanmıştır. Etmenlerin ürettiği bilgilerle doldurulan küp üzerinde, iyi bilinen madencilik tekniği ilişki kuralı çıkarımı tekniği uygulanmıştır. İkinci olarak; yeni bir modelle çok seviyeli ilişki çıkarımı aynı küpe uygulanarak sonuçlar karşılaştırılmıştır [Kaya ve Alhajj, 2005].

Alkharouf ve ark. (2005), küçük boyutlu bir veritabanında, OLAP ve küp mimarisi kullanarak kümeleme çalışması yapmışlardır. Kendi önerdikleri yaklaşımı, Kmeans ve SOM ile kıyaslanarak, OLAP'ın hızı ve küp yapısındaki işlemlerin yararlığı ortaya konmuştur [Alkharouf ve ark., 2005].

Ticari ilişkisel VT sistemleri karmaşık iş modellemelerine istenilen desteği verememektedir. ANSISQL, ilişkileri çok boyutlu diziler gibi ele almamaktadır.İş modellemesi birbiri ile ilişkili formuller üzerinde operasyonları gerektirmektedir. Bu sebeple uygulama desteği verebilen ilişkisel OLAP (ROLAP)'ın ve çokboyutlu OLAP (MOLAP)'ın kullanımı üzerinde durulmuştur. Witkowski ve ark.(2005) SQL uzantısı olan ilişkiler üzerindeki karmaşık modellemede, dizi hesaplamalarına imkan sağlayan hesap çizelgesi (Pivot Table) hazırlamıştır. Bütünleşik operasyonlardaki

derleme işleminde zaman verimliliğine dikkat çekilerek; erişim yapıları ve uzantı modellerdeki iyileştirme sunulmuştur [Witkowski ve ark., 2005].

Shi ve ark. (2007), KDS geliştirme platformu olan MSMiner'ın tasarımı ve uygulamasından bahsetmiştir. Sistem bir VA üzerinde inşa edilmiştir ve bazı VM algoritmaları ile entegre edilmiştir[Shi ve ark., 2007].

4.2. Önerilen Model: Bütünleşik Veri Kübü Sistemi (BVKS)

Günümüze kadar sadece raporlama amaçlı kullanılan veri küplerinin hem hız hem de fonksiyonellik avantajı olduğu birçok araştırma tarafından gösterilmiştir[Han ve Kamber, 2001]. Ancak şimdiye kadar bir veri küpündeki bilgilerin dışsal bir karar modeline veri olarak kullanıldığı ve bunlardan yeni veri küpleri ya da raporların elde edildiği bir çalışmaya literatürde rastlanmamıştır. Bu çalışmada BVKS olarak adlandırılan yapı, OLAP küplerinin analizi aşamasında veri küplerinin hız ve fonksiyonellik avantajlarını kullanarak, amaca bağlı olarak analitik modelleri ve paket programları da dahil ederek, küp üzerinde basit işlemler ve hesaplamalar ile yeni küplerin ve raporların türetilmesine imkân vermektedir.

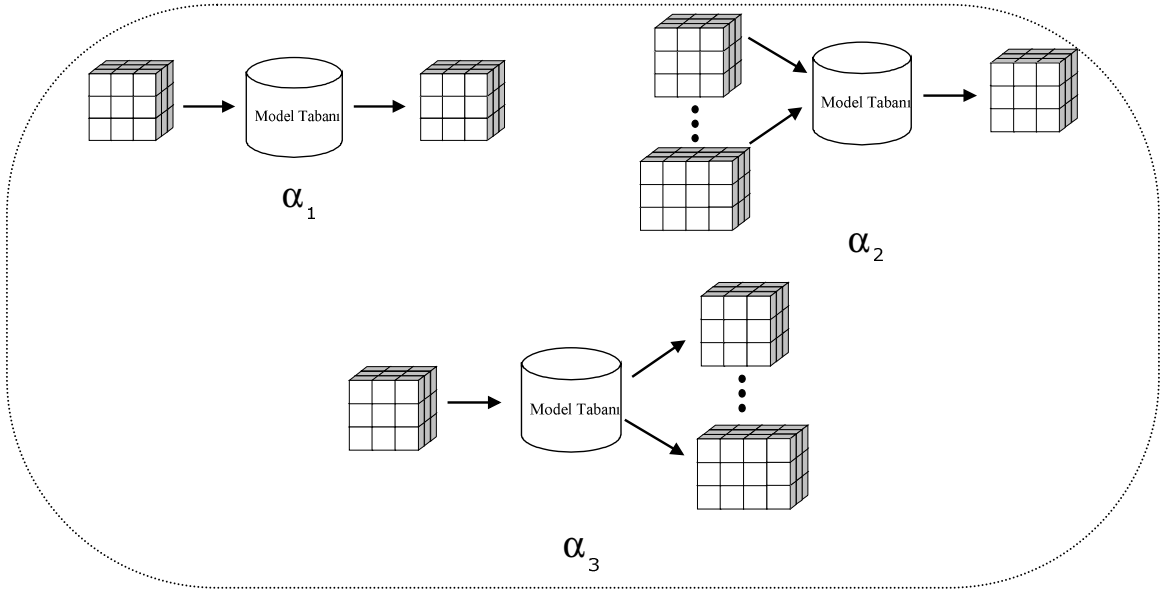
Büyük hacimli veri tabanlarında işlem yapma ve sorgu sonuçlarını almak oldukça yoğun işlem gerektiren ve zaman açısından memnuniyet verici sonuçları olmayan bir süreçtir. Küp oluşturulması ve özet bilgilerin kullanılmasıyla bu sıkıntı kısmen aşılmaktadır. BVKS mevcut OLAP mimarisine ek olarak, veri küpleri üzerinde etkin ve esnek çalışma imkânı sağlayacaktır. Önerilen yaklaşımda BVKS ve çevre birimleri (OLAP, rapor, analitik model vb.) arasındaki ilişkiler göz önüne alınarak aşağıdaki gibi bir sınıflandırma yapılabilir.

- α : Sadece veri küplerinin kullanıldığı yapıdır. Modül içerisinde kodlaması yapılmış analitik algoritmalar (SQL komutları vb.) amaca yönelik olarak yeni küp yapısını oluşturacaktır.
- β : Küplere ek olarak, sisteme harici bir girdi ya da çıktı (analitik rapor, form vb.) dâhil edilmesi ile ortaya çıkan yapıdır.

- σ : Veri küpü sisteminde paket programların (istatistiksel, optimizasyon, simülasyon vb. paket programlar) araç olarak kullanıldığı yapıdır.
- γ : α , β , σ durumlarının eş zamanlı kullanıldığı geniş ve karmaşık entegre yapıdır.

OLAP kullanılarak elde edilen küpler dinamik dizi ya da objelere aktarılacaktır. Böylece araçların standart işlemlerinin (slice and dice, drill down, pivoting vb.) dışında, veri küplerinin kullanımı sağlanacaktır. Şekil 4.1'de α , Şekil 4.2'de β ve Şekil 4.3'de σ yapısına örnek verilmektedir. Şekil 4.4 ise bütün durumların eş zamanlı dikkate alındığı entegre yapı gösterilmektedir. Bu yapılarda Model Tabanı esas olarak aşağıda listelenen başlıkları içermektedir [Han ve Kamber, 2001].

- Veri tabanı teknolojisi
- Bilgi sistemleri
- İstatistik
- Makina öğrenimi
- Görselleştirme ve diğer disiplinler

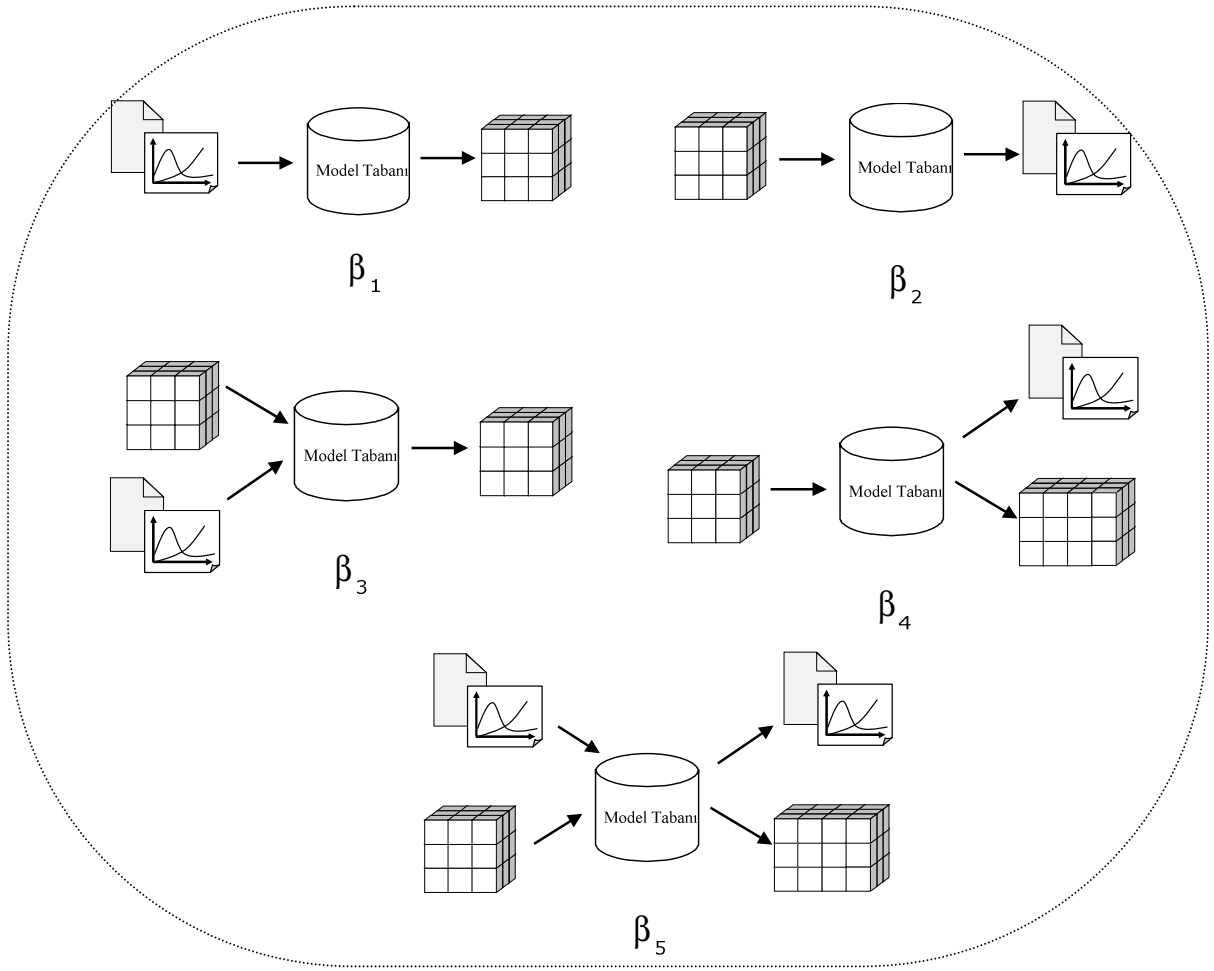


Şekil 4. 1. Önerilen modelin getirdiği yeni α küp işlem senaryoları
 α_1 : Bir küpten yeni bir küpün elde edilmesi
 α_2 : Birden fazla küpten yeni bir küpün elde edilmesi
 α_3 : Bir küpten birden fazla yeni küpün elde edilmesi

α_1 uygulaması, kullanıcı tarafından doğrudan ya da başka bir ortamdan veri transferiyle küp ortamına kaydedilen ve saklanan veriler, model tabanında çalışacak model için girdi oluşturacaktır (örneğin, model tabanında çalışacak bir regresyon modelinin katsayıları bu küpten alınabilir). Model çalıştırılması sonucunda elde edilen sonuç verileri, ya başka bir küp uygulamasında kullanılmak ya da doğrudan raporlama amaçlı kullanılmak üzere diğer küpte saklanacaktır. α_2 uygulamasında, model tabanında çalışacak model için birden fazla küpten veri çekilmektedir (örneğin, malzeme ihtiyaç planlamasındaki üretim emri ya da satınalma emri hesaplamalarını yapacak bir model için, ana üretim çizelgesi (MPS) değerlerinin bir küpten, malzeme listesi (BOM) değerlerinin ise diğer küpten çekilmesi). Elde edilen sonuç verileri, α_1 uygulamasında olduğu gibi başka bir küpte saklanabilmektedir. α_3 uygulamasında ise, tek bir küpten veri isteyen model (model tabanından), sonuçlarını birden fazla küpte saklayabilmektedir. Bu durum aslında, sonuçların başka uygulamalarda kullanılmak üzere tasnif edilmesi olarak yorumlanabilir.

β uygulaması, α uygulamasına ek olarak raporların(karakter döküman, hesap tablosu, XML dosyalar, vb.) işleme dahil edildiği durumdur. Kullanıcı tarafından doğrudan ya da başka bir ortamdan veri transferiyle küp ortamına kaydedilen ve saklanan verilerin haricinde raporlar, model tabanına girdi ya da çıktı olarak kullanılmaktadır. β_1 ve β_3 işlemleri raporların girdi olduğu durumu göstermektedir. β_2 ve β_4 işlemleri raporların çıktı olduğu durumu göstermektedir. β_1 sadece raporun girdi olduğu durumdur. β_1 durumuna örnek olarak, İMKB’de hisse senedi bazında göstergelerle teknik analiz yapılırken 30 ve 100 indekslerinin eğilim raporunun girdi olarak kullanılması sayılabilir. Tüm hisse senetlerinin verilerinin kayıtlı olduğu bir küp ve kıyaslama için girdi olarak kullanılan bir rapor modele dahil edildiğinde, birden fazla hisse senedinin analizi aynı anda gerçekleştirebilen ve sonuçları küpe kaydeden β_3 durumu elde edilecektir. β_2 , kullanıcı tarafından oluşturulmuş bir küpün ya da sistemde mevcut bir küpün girdi olarak kullanıldığı, çıktısının başka bir modele girdi ya da yöneticiye karar desteği veren bir rapor olarak sunulduğu işlem senaryosunu göstermektedir. β_4 ise girdi olan küpten amaca bağlı olarak yeni küplerin ve raporların elde edildiği durumu göstermektedir. β_5 , girdisi ve çıkışı küp ve rapordan oluşan durumdur. Sistemde mevcut olan ya da kullanıcı tarafından yeni oluşturulan

bir küp(örneğin, büyük bir marketle ilgili satış rakamlarını içinde barındıran bir küp) ve harici bir takım raporlar(örneğin, sektörel bazda tedarikçi firma bilgileri) β_5 durumu için girdi oluşturmaktadır. Çıktısı ise yine mevcut küpün analitik işlemlerle dönüştürülmüş hali ya da yeni oluşturulmuş bir küp(örneğin, satış tahmini sonucu hesaplanan değerleri saklayan tahmin küpü) ve rapor dosyalarıdır(örneğin, performansı nedeni ile çalışılma kapasitesi değiştirilecek tedarikçi firmalar).



Şekil 4. 2. Önerilen modelin getirdiği yeni β küp işlem senaryoları

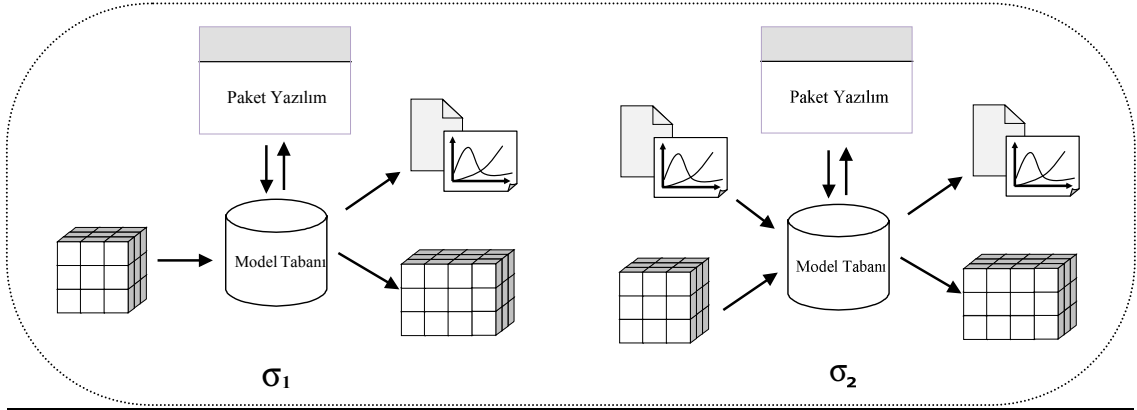
β_1 : Harici girdi kullanılarak yeni küp(ler) elde edilmesi

β_2 : Küp(ler) kullanılarak harici çıktının elde edilmesi

β_3 : Küp(ler) ve harici girdi kullanılarak yeni küp(ler) elde edilmesi

β_4 : Küp(ler) kullanılarak yeni küp(ler) ve harici çıktının elde edilmesi

β_5 : Küp(ler) ve harici girdi kullanılarak yeni küp(ler) ve harici çıktının elde edilmesi

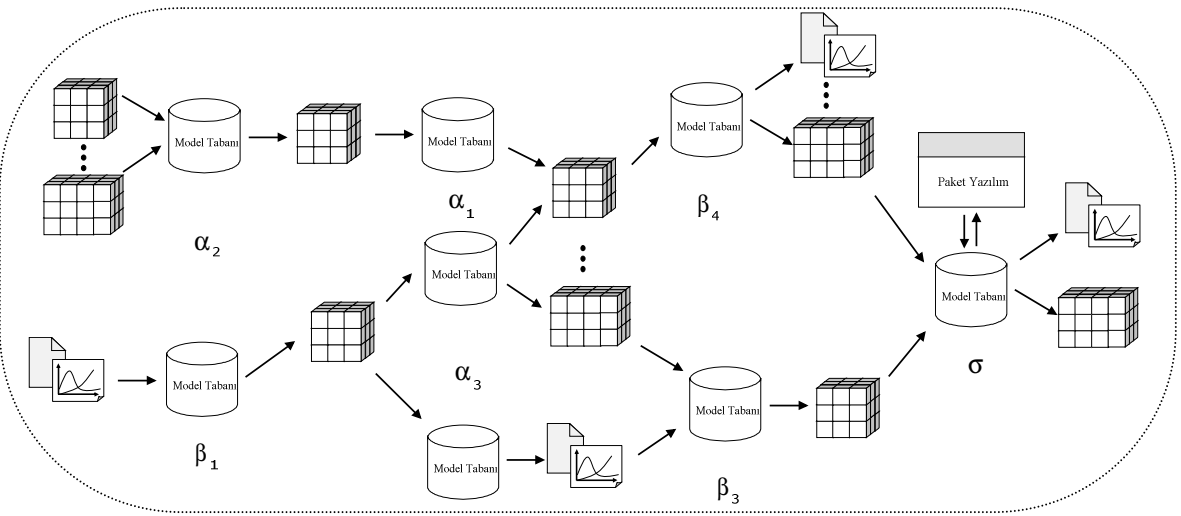


Şekil 4. 3. Önerilen modelin getirdiği yeni σ küp işlem senaryosu

σ_1 : Küp(ler) kullanılarak paket programlar (istatistiksel, optimizasyon simülasyon vb. paket programlar) vasıtasıyla yeni küp(ler) ve harici çıktının elde edilmesi.

σ_2 : Küp(ler) ve harici girdi kullanılarak paket programlar (istatistiksel, optimizasyon, simülasyon vb. paket programlar) vasıtasıyla yeni küp(ler) elde edilmesi.

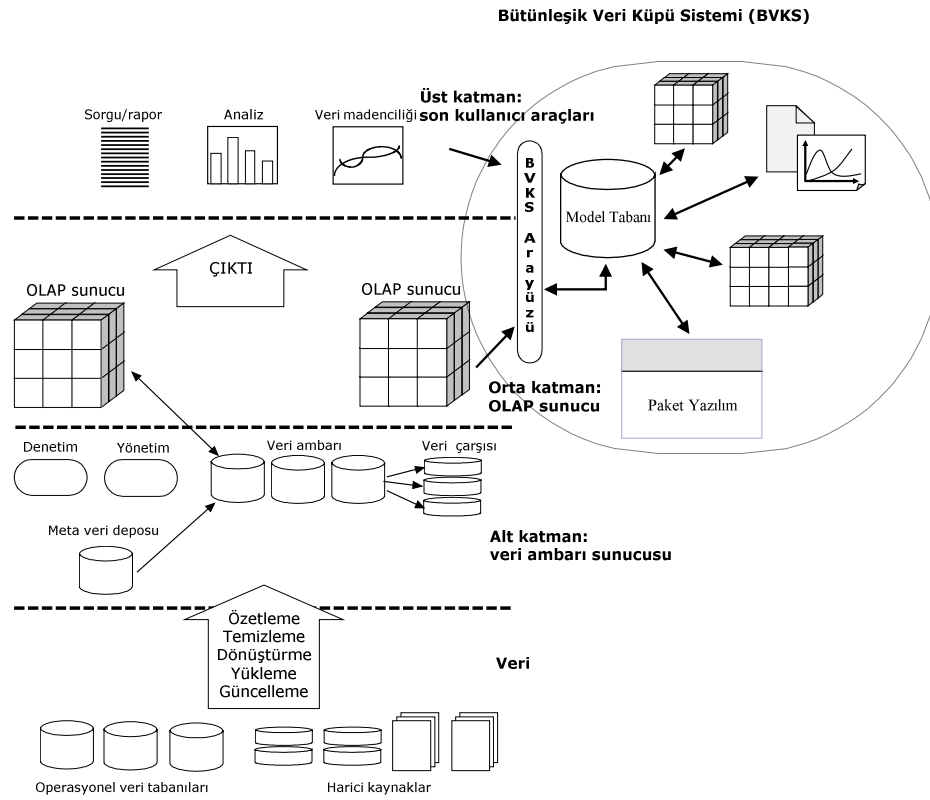
σ_1 ve σ_2 uygulaması, paket yazılımların (SPSS, Gams,Arena vb.) sisteme dahil edildiği durumlardır. Amaca ve paket yazılıma bağlı olarak model tabanı girdisi bir küp ya da küple birlikte rapor dosyaları olabilir. Önerilen mimarinin testi kısmında σ durumunu da kapsayan bir satış küpü uygulaması yapılmıştır.



Şekil 4. 4. Önerilen modelin getirdiği yeni γ (entegre yapı) küp işlem senaryoları

Mevcut satış kayıtlarını saklayan bir küp kullanıcı tarafından oluşturulmuştur. Bu küpte saklanan veriler, SPSS yazılımına regresyon denklemi katsayısı hesabı için girdi oluşturmuştur. Katsayılarla elde edilen regresyon denklem ile yapılan satış tahminleri yeni küpe yazılmıştır. Regresyon denklemi güvenilirliği ile ilgili raporlarda çıktı olarak σ durumu uygulamasında yer almıştır.

γ (entegre yapı) küp uygulaması, tüm α , β , ve σ durumlarını içeren işlem senaryosunu göstermektedir. Pratikte, tüm uygulamaları kapsayan entegre yapı ile çok nadir karşılaşılır.



Şekil 4. 5. BVKS ile elde edilen yeni 3 katmanlı VA mimarisi

Tüm OLAP yapılarına entegre edilebilir olan önerdiğimiz mimari OLAP uzantısı olarak düşünülmüştür. Veri depoların işlenmesi ile elde edilen VA, VK için kaynak oluşturmaktadır. OLAP'ın avantajlarını kullanarak işletilmesi düşünen sistem için VK temel veri kümesini oluşturmaktadır. Genel OLAP yapısından elde edilen ve tüm

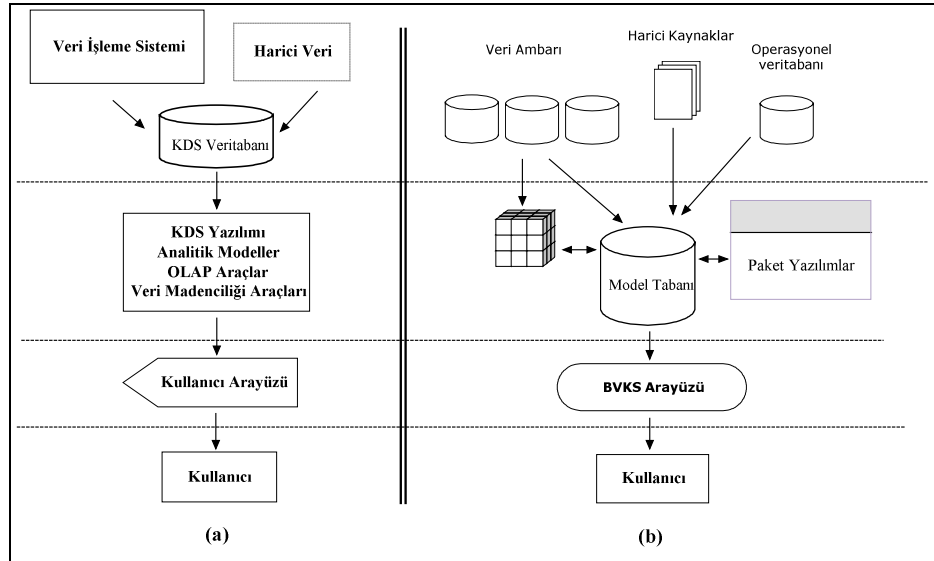
seviyeleri barındıran küpler; nesne ya da dinamik dizilere aktarılarak kullanım kolaylığı sağlanmıştır. Yeni küpler OLAP ın tüm işlemlerini (drilldown, rollup, slicing ve dicing etc.) yapabilmeye imkân sağlamaktadır. Bütünleşik veri küpü sistemi ile amaca yönelik olarak kullanılacak modeller, küpler ve harici verilerle sistem bütünleşmektedir. Bahsi geçen girdiler ve çeşitli raporlar aynı zamanda sistemin çıktısında olabilmektedir. Şekil 4.5 üç katmanlı VA mimarisi [Han and Kamber,2001] ve BVKS nin bütünleşik yapısını göstermektedir.

4.3. BVKS'nin Karar Destek Sistemi Yapısı

Karar destek sistemleri (KDS) bir bilgi sistemi olup, özellikle belirsizlik seviyesi yüksek olan kararlar için analitik model(ler) kullanarak karar vericiye destek sağlayan sistemler olarak tanımlanmaktadır. Yöneticilerin, yönetsel problemlerin aşılması için kantitatif modelleri kullanma çabalarıyla ortaya çıkan karar desteği, ilk J. D. Little (1970) ın çalışmasıyla ortaya konmuştur. Terim olarak KDS in kullanıldığı ilk çalışma ise Gorry ve Scott Morton (1971)'a aittir [Marakas ,1999].

İki tip KDS vardır : Model-odaklı ve veri-odaklı [Dhar ve Stein, 1997]. Model-odaklı KDS herşeyden önce, “What if” ve diğer farklı analizlerin yapılması için bazı modeller kullanan büyük organizasyonel bilgi sistemlerinden bağımsız, tek başına sistemlerdir. Bu gibi sistemler genellikle merkezi bilgi sistemi kontrolü altında olmayan son kullanıcı bölümler ya da gruplar tarafından geliştirilirler. Bu sistemlerin analiz yetenekleri, modelin kullanımını kolaylaştıracak iyi bir kullanıcı arayüzüyle birleştirilmesine bağlıdır. Veri-odaklı KDS ler, büyük organizasyonel sistemlerde bulunan büyük veri havuzlarını analiz eden sistemleridir. Bu sistemler, daha önceden büyük miktarlardaki verilerde saklı kalan faydalı bilgilerin çıkarılarak, kullanıcılara karar verme desteği sağlayan sistemlerdir. Veri işleme sistemlerinden elde edilen veriler, bu amaç için genellikle veri deposunda toplanırlar [Laudon and Laudon, 2002]. BVKS, büyük hacimli VT ve VA'ndan analiz ve keşfe yönelik olarak veri işleme yapan ve kısmen model çalıştıran veri odaklı bir KDS 'dir.

Karar destek sistemlerinin ana bileşenleri, KDS veritabanı, KDS yazılım sistemi ve kullanıcı arayüzüdür. KDS veritabanı, bir kişisel bilgisayara yerleştirilecek kadar küçük bir veritabanı ya da çok büyük veri deposu şeklinde olabilir. Bir karar destek sisteminin bileşenleri ve BVKS'nin KDS yapısı Şekil 4.6 'da verilmektedir. BVKS açısından KDS bileşenlerini; KDS veritabanını oluşturan Operasyonel VT ve VA, harici veriler ve kullanıcı tarafından girilen veriler, KDS analitik model tabanını oluşturan standart OLAP işlemleri, paket programlar ve analitik modeller, BVKS arayüzü ve kullanıcı oluşturmaktadır.



Şekil 4. 6. a) Temel KDS bileşenleri [Marakas, 1999] b) BVKS'nin KDS bileşenleri

5. BÜTÜNLEŞİK VERİ KÜPÜ SİSTEMİ (BVKS) UYGULAMASI

5.1. Bütünleşik Küp Modulu SATIŞ KÜPÜ Uygulaması

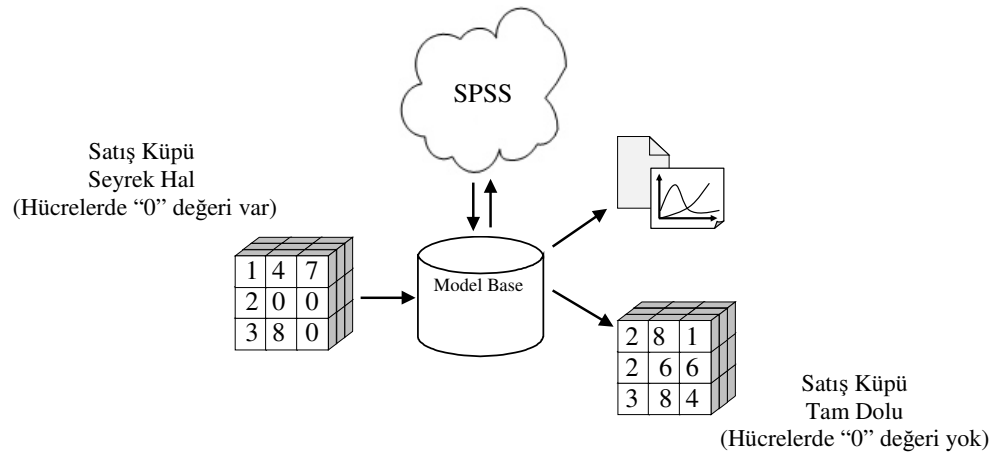
OLAP küplerinin analizi aşamasında küp üzerinde basit işlemlerin, hesapların yapılarak yeni küplerin türetilmesinin şu an mümkün olduğu bilinmektedir. Büyük hacimli veri tabanlarında işlem yapma ve sorgu sonuçlarını almak oldukça yoğun işlem gerektiren ve zaman açısından memnuniyet verici sonuçları olmayan bir süreçtir. Küp oluşturulması ve özet bilgilerin kullanılmasıyla bu sıkıntı kısmen aşılabacaktır. Şekil 5.1’de işaretlenmiş alan içinde bulunan modülün OLAP mimarisindeki katmanlar arasına eklenmesi ile, küplerin türetilmesi aşamasında, dışardan farklı dosyalardan ya da çeşitli küplerden gelen verilerin sisteme girdi olmasıyla yeni küpler elde edilebilecektir. Bu yaklaşımda, bileşenler kullanılarak oluşturulan küp sistemi, analitik algoritmalarla desteklenen bir modül ile daha iyi sonuçlar veren bir yapıyı sunacaktır. Önerilen mimarinin uygulanabilirliğini göstermek için küçük bir uygulama gerçekleştirilmiştir. SQL Server 2000 kullanılarak Northwind satış veri tabanından satış veri küpü oluşturulmuştur. Oluşturulan veri küpünde boyutlar(dimension); zaman, ürün ve lokasyon, hesaplama (measure) değeri ise satış toplamaları bilgisidir. Küpteki veriler küp yapısına uygun olan çokboyutlu dizilere aktarılmıştır. Daha sonra bu mimarideki veriler kullanılarak ürün ve lokasyonlar bazında regresyon modelleri oluşturulmuştur. Bu modeller kullanılarak yapılan tahminler ise yine küp mimarisine uygun olarak oluşturulmuş dinamik dizilere aktarılmıştır. Elde edilen dinamik dizilerin daha sonra OLAP veritabanına aktarılmasıyla, bütünleşik küp sistemi çevrimini tamamlamış olacaktır.

OLAP yapısına eklenebileceği düşünülen Küp Modülünün kompozit yapısı Şekil 5.1’de gösterilmektedir. Mevcut OLAP mimarisinde; VA’ndan kullanıcının amacına yönelik olarak bileşenler seçilmekte ve Küp oluşturulabilmektedir. İstenilen özet bilgiye çok hızlı bir şekilde sorgular yardımı ile ulaşılabilmektedir. Önerilen yapı ile Küp mantığı kullanılarak ve onun faydaları göz önüne alınarak; mevcut VA’nı kullanarak ya da VT’nı işleyip kullanarak farklı küplerin oluşturulabileceği, bu küplere farklı dosyalardan (.doc, .xml, .txt, ...) bilgi aktarıp yeni küplerin

oluşturulabileceği bir senaryo ortaya konmaktadır. Küpler birbirleri arasında ve farklı dosyalarla ilişki içinde yapısını değiştirebilmektedir. Farklı amaçlarla hazırlanmış yazılımlara (GAMS, ARENA vb.) girdi sağlama yeteneğine sahip küplerin kullanımıyla elde edilecek program çıktıları, ileride kullanılacak yeni yapılara veri girişi imkanı sağlayacaktır.

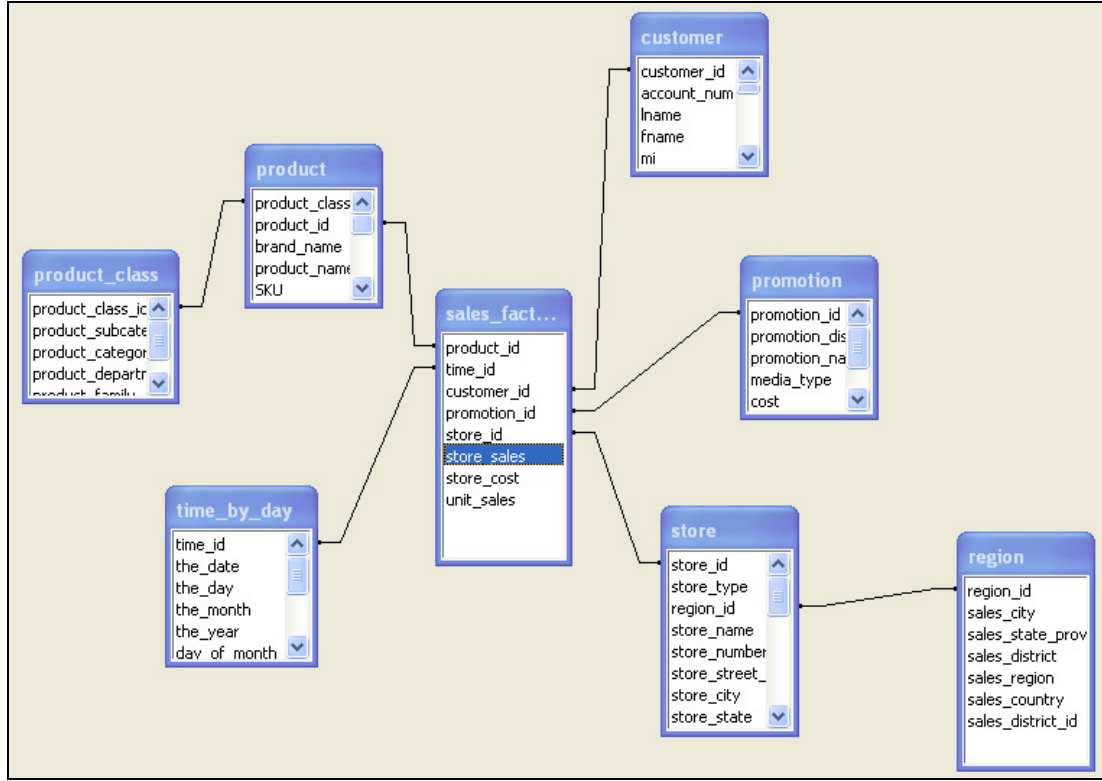
Bütünleşik küp modülü yapısında, VA'nı kullanan OLAP Server, Veri hazırlama kısmında VTYS desteğinde, veri temizleme, indirgeme ve dönüştürme işlemleri yapan kısım, daha önceden oluşturulmuş veri küpleri ve paket ya da özel yazılımlar yer almaktadır. Tüm veri girişi ve çıkışlarının ortasında karar desteği sağlayacak, veriyi bilgiye dönüştürecek tüm VM tekniklerini kapsayan kısım yer almaktadır.

Yapılan uygulama ile önerilen BVKS'nde Şekil 5.1'de gösterilen; σ yapısı elde edilmiştir.



Şekil 5. 1. Önerilen mimari ile paket yazılım ve küp girdisiyle, rapor ve yeni küpün oluşumunun uygulamadaki yapısı

OLAP çalışmalarında genelde kullanılan Northwind veri tabanı, uygulama için test verisi olarak kullanılmıştır. OLAP yapısına modulun yeni bir seviye olarak girebileceği düşünüldüğünde, katman olarak girmesi mümkündür. Şekil 5.2'de VT'nın tablo formatındaki görüntüsü verilmiştir.

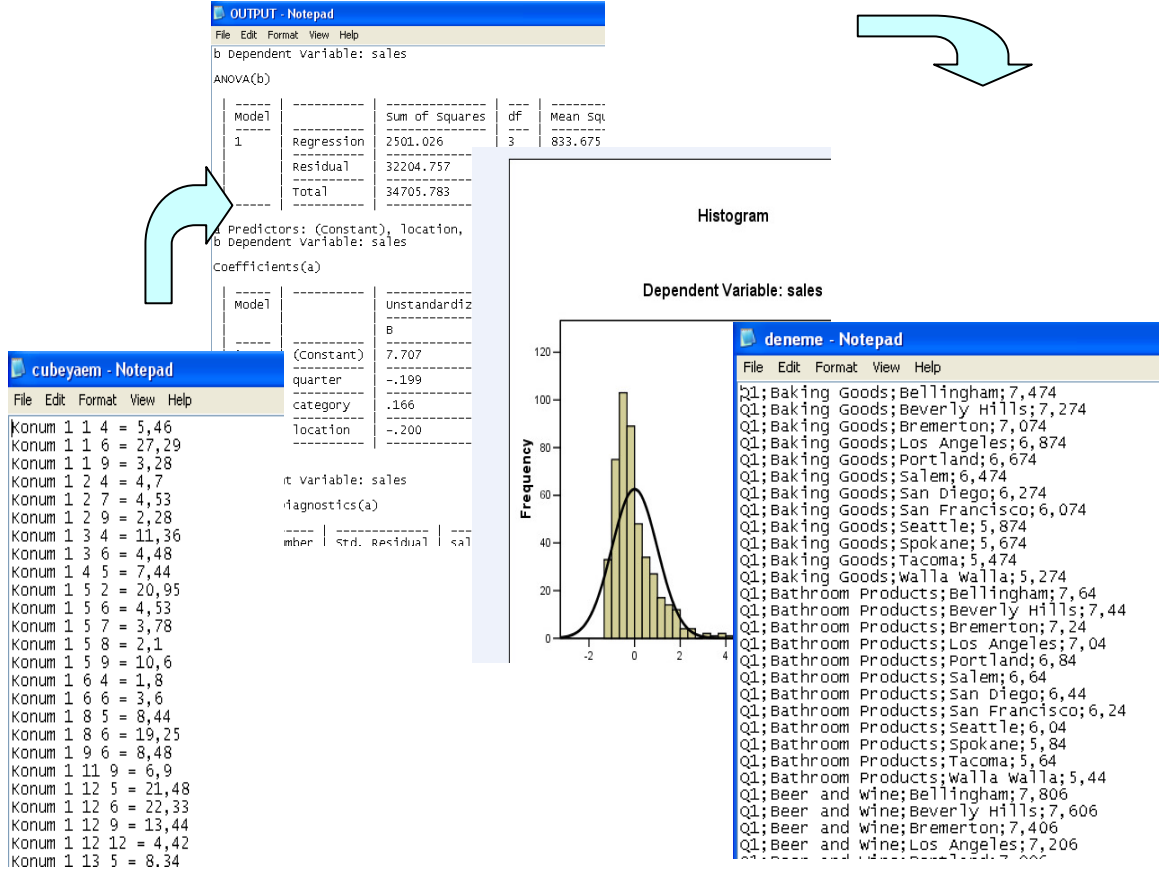


Şekil 5. 2. Küpü oluşturan boyut tabloları

Şekil 5.2’de .NET 2005 C# dili kullanarak kodlanan uygulama arayüzü görülmektedir. VA’nı kullanarak sunucudan elde edilen satış küpü, boyutları ve hesaplamaları değişmeyecek şekilde kodlama dilinde dinamik dizilere ya da objelere aktarılmıştır.

3 boyutlu olarak çalışılan küp, çeyrek yıl (quarter), ürün kategorisi (product_category) ve lokasyon (store_city) olarak oluşturulmuştur. Şekil 5.3’ de OLAP sunucu da kullanılan VA’ndan elde edilen Satış Küpünün dizilere aktarımı ve yeni küpün oluşturulması görülmektedir. Elde edilen dizi yapılı küp, girdi olarak kullanılmıştır ve SPSS yazılımı kullanılarak bir regresyon denklemi için katsayılar elde edilmiştir. Bu verilerden hareketle elde edilen regresyon denkleminde boyutlar, bağımsız değişkenleri oluşturmuştur. Satış toplamaları ise bağımlı değişkenlerdir.

özetleme ve sorgulama hızı bir araya geldiğinde, çok güçlü bir Karar Destek Sisteminin ortaya çıkması olağandır.



Şekil 5. 4. Tahmine yönelik regresyon modeli uygulaması

Satış küpü modülünde elde edilen küp 3 boyutla tariflenmiştir. 1. boyutta Yıl 4 Çeyrek'ten oluşmaktadır. Q1: 1. çeyrek yıl (quarter1); Q2: 2. çeyrek yıl (quarter2); Q3: 3. çeyrek yıl (quarter3); Q4: 4. çeyrek yıl (quarter4) ı simgelemektedir. 3 aylık toplam da satışları göstermektedir. 2. boyutta ise 44 tipden oluşan ürün kategorisi yer almaktadır. Lokasyon boyutunda 12 adet şehir yer almaktadır. Küp 4*44*12 boyutlu olduğundan dolayı, 2112 hücreye sahiptir. Elimizdeki veri ambarında bu üç boyutu kapsayan 721 adet kayıt mevcuttur. Bazı çeyrek yıllarda, bazı ürün kategorisi için çeşitli şehirlerde satış yapılmamış olmamasından dolayı kayıt yoktur. Bu gibi durumlarda küpün boyutları ile ilgili alanlar boş geçer ve hesaplama değeri 0 olarak değerlendirilir.

Elde edilen regresyon denkleminin kullanılması ile yeni küpün 2112 hücresi de dolu olarak elde edilmiştir. Veri tabanı ve ambarı yönetim sistemlerinde Veri Madenciliğine hazırlık aşaması olan veri işleme kısmında bu yöntem kullanılarak veri tabanındaki gürültü (noisy) azaltılabilir ve VM tekniği öncesi veri eksikliği olmayan bir yapı elde edilebilir.

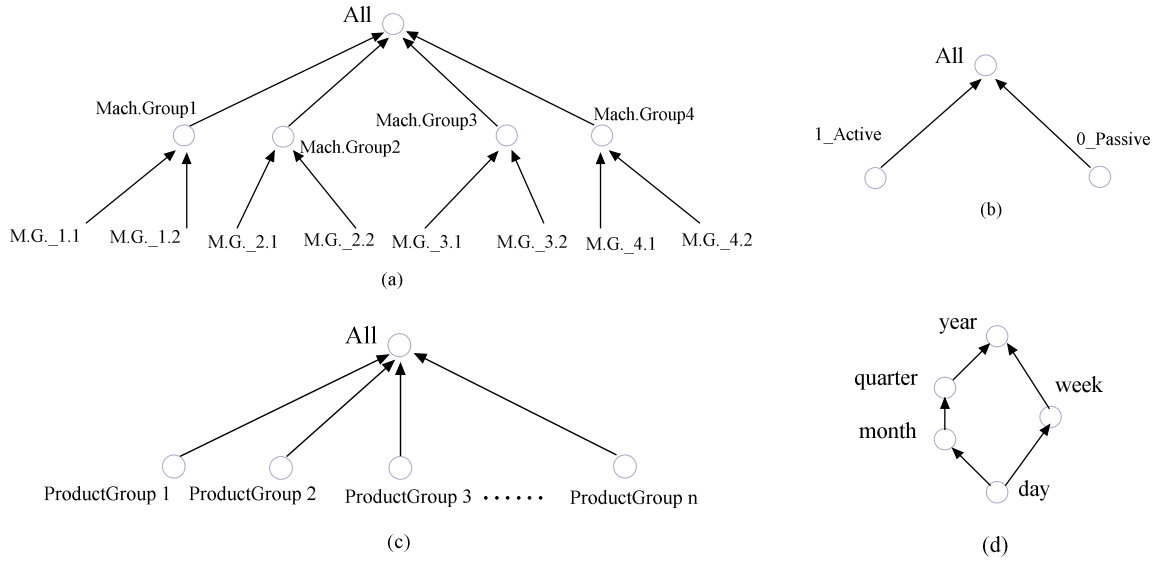
5.2. BVKS odaklı OLAP ve VTYS Performans Uygulaması

Yeni mimarinin performansını ortaya koyabilmek için BVKS odaklı bir uygulama yapılmıştır. İlk kısımda ilişkisel veritabanı üzerine ve sonrasında aynı yapı OLAP-VK üzerinde uygulanarak sonuçlar ortaya konmuştur. BVKS'nin VK'nü kullanım avantajı yanısıra, VK'ne bağlı olarak sorgu işleme ve raporlama performansı çok yüksektir. Uygulama için kullanılan VT, halen faaliyetlerine devam eden ve imalat sektöründe yer alan bir firmaya aittir. 07/2002-06/2003 zaman aralığına ait olan VT büyüklüğü 8 GB dır. Firma VT, üretim ile ilgili veri yapılarını barındıran bir organizasyonel VT dır. 105 adet tabloya sahip VT, bazı tablolardaki kayıt sayısı 1.3×10^6 civarındadır. Stok hareketleri dosyası 3×10^6 kayıta kadar çıkmaktadır.

Firma, veri tabanı yönetim sistemi üzerine kurulu bir bilgi sistemi kullanmaktadır. Bilgi sistemi olmasına rağmen, KDS eksikliği nedeni ile sipariş üzerine çalışan firmada, gelen taleplere termin verilmesi aşamasında çok sıkıntı yaşanmaktadır. Mevcut haliyle sipariş sorumlusu sezgilerine dayalı olarak teslim zamanı bildirmekte ve aksamalara, memnuniyetsizliğe yol açmaktadır. Bu sıkıntının çözümü için, FIFO yaklaşımı kullanılarak bir çizelgeleme uygulaması yapılmıştır. Uygulama arayüzlerinden bazıları EK1 de verilmiştir.

BVKS için oluşturulan küp te kullanılan merkezi tablo (fact table) üretim tablosudur. Boyutlarımız ürün, status (işin aktiflik durumu), makinalar ve zaman boyutudur (Bkz. Şekil 5.5). Hazırlanan yapı, kar tanesi (Snowflake) şemasına uygun olarak tasarlanmıştır. Üretim küpünün altyapısı hazırlandıktan sonra, standart küp işlemleri SQLServer2000 Enterprise Manager kullanılarak prosedür, fonksiyon ve trigger yapısı oluşturularak elde edilmiştir. Uygulamada kullanılan üretim küpü boyut hiyerarşisi

Şekil 5.5’ de verilmiştir. Makina boyutunda ALL tüm makinaları kapsamaktadır. Makina grupları (Mach.Group1, Mach.Group2, Mach.Group3, Mach.Group4) problemde sürece göre oluşturulmuştur. Alt gruplar ise (M.G.1.1, ... , M.G.4.2), alt süreçlere göre gruplanmıştır. Status boyutu ise makina ya da ürün açısından uygunluk durumunu göz önüne almaktadır. Atamalar durumu 1 olan makinalar için gerçekleştirilmektedir. Ürün gruplarında, yarı mamulun izleyeceği sürece göre oluşturulmuştur. Ürün tiplerine göre, yarı mamulun izleyeceği süreç ve sırası değişmektedir. Zaman boyutunda günlük, haftalık, aylık, çeyrek yıllık ve yıllık periyotlar kullanılmıştır.



Şekil 5. 5. Üretim küpü boyut hiyerarşisi (a) Makina boyutu (b) Status boyutu
(c) Ürün boyutu (d) Zaman boyutu

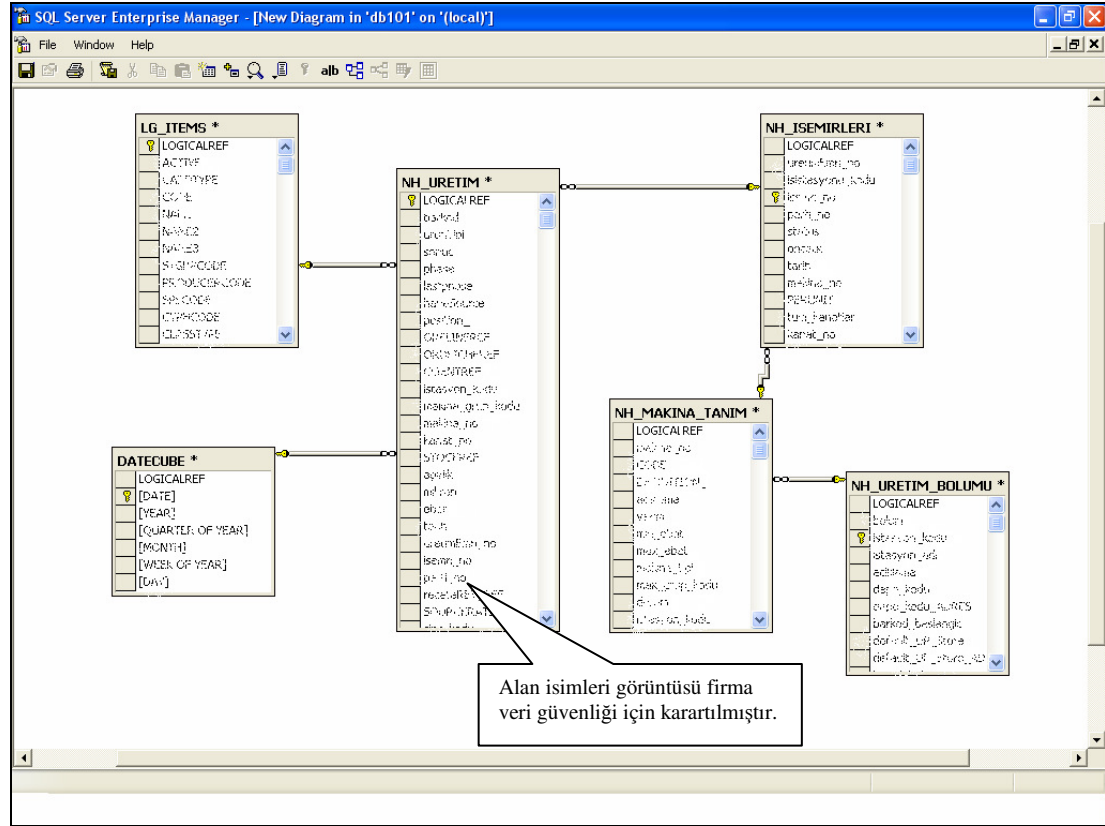
Önerilen mimarinin uygulanabilirliği ve avantajının görülebilmesi için bir firmada BVKS yapısı kurulmuştur. Firmanın problemi müşterilere termin zamanını bildirememesinden kaynaklanmaktadır. Bu sıkıntının aşılması için ilk olarak mevcut veri yapısı üzerinde bir uygulama gerçekleştirilmiştir. İlişkisel tablolar kullanan sisteminde bazı tablolar $3 \cdot 10^6$ kayıta kadar çıkmaktadır. Günlük işlem hacmi fazla olduğu için her geçen gün de büyümektedir. Termin verilebilmesi için, problem çizelgeleme problemi olarak ele alınmıştır. Çözüm için ihtiyaç duyduğumuz girdiler, ürünler, üretim zamanları, mevcut ve uygun makinalar, fire miktarları ve işe

başlayabilme zamanıdır. Süreç analizi neticesinde karma bir yapının olduğu görülmüştür. Ürün tipine göre izlenen süreç değişmektedir. Benzer şekilde birim üretim zamanı ve fireler de değişkenlik göstermektedir. Bu durumda her sipariş için, sürece bağlı kalarak, üretim zamanı ve fireler her bir makina için hesaplanmalı ve daha sonra ataması yapılmalıdır. Aynı ürünün diğer makinaya atanmasında da aynı yaklaşım kullanılarak bu işlem süreç tamamlanana kadar devam ettirilmelidir. Bu şekilde gerçekçi bir termin zamanı elde edilip müşteriye bildirilebilecektir.

İlk uygulama, ilişkisel tablolar üzerinde kurulu VTYS tabanlı bir yaklaşım kullanılarak gerçekleştirilmiştir. Girdiler için gerekli tablolar kullanılarak hesaplamalar yapılmış ve atamalar yapılarak bir çizelge ortaya konmuştur. Termin zamanları ile ilgili elde edilen sonuçlar uygun olmasına karşın uygulamanın sonuçlanması Çizelge 5.1 - Çizelge 5.3 'de görüldüğü gibi uzun zaman almaktadır. Bunun sebebi VT üzerinde çalışan sorgulama hızından kaynaklanmaktadır. Bu firma ve müşteri tarafından kabul edilemez ve tatmin etmeyen bir durumdur. Termin zamanı firma için müşteri ile temas anında verilmesi gereken bir bilgidir. Bundan dolayı siparişi tamamen öngörü ve sezgilerine dayanarak bilgi vermektedir.

Sonraki uygulamamızda ise, daha önceki çalışmalarda rastlamadığımız BVKS yaklaşımı ile problem tekrar çözülmüştür. Uygulama ile sırasıyla Şekil 4.1 ve Şekil 4.2' de gösterilen α ve β yapıları elde edilmiştir. Şekil 4.3'de yer alan σ yapısı ise Çetinyokuş ve Gökçen (2008) çalışmasında yer almıştır. Öncelikli olarak ilişkisel VT üzerinde VA oluşturulmuştur. Daha sonra VA kullanılarak amaca yönelik olarak 4 boyutlu Üretim Küpü elde edilmiştir. Üretim Küpünün, Ürün boyutunda ürünün koduna bağlı olarak süreçler elde edilmiştir. Makina boyutunda ise süreçlere göre makina grupları yer almaktadır. Makina durumlarını gösteren Active ve Passive bir diğer boyutumuzdur. Makinaların birim üretim zamanları ve fire oranları geçmiş verilerden yararlanılarak sistem üzerinde alt hesaplamalar olarak karşımıza çıkmaktadır. Zaman boyutu bizim ne kadarlık bir periyotda geriye giderek hesaplama için kullanacağımızla ilgilidir. BVKS ile elde edilen sonuçlar Çizelge 5.1 - Çizelge 5.3'de incelendiğinde yeni yaklaşımın üstünlüğü ortaya çıkmaktadır. Aynı platformda yapılan denemelerin neticesinde BVKS yaklaşımının ilişkisel yaklaşıma

göre çok üstün olduğu görülmüştür. Uygulama 4H..... ürünü için 400 birimlik sipariş miktarı ve son 1 yılın verileri kullanılarak uygulama çalıştırıldığında, VTYS 3691883 ms., BVKS 1459 ms. CPU zamanında gerçekleşmiştir. Önerile yapının süresi, VTYS'nin ikibinbeşyüzde biridir.

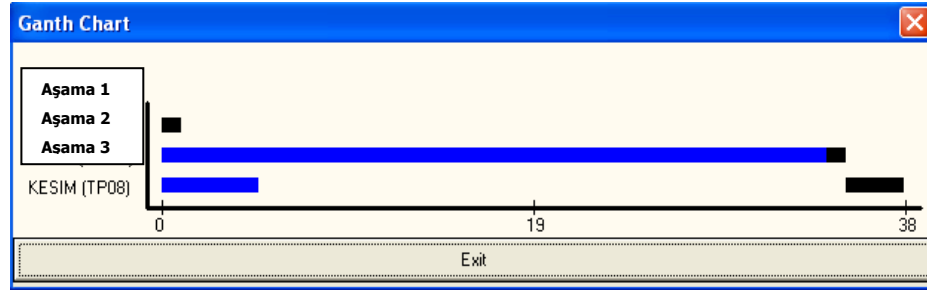


Şekil 5. 6. BVKS çizelgeleme uygulaması için kullanılan VT şeması

BVKS için oluşturulan küp te kullanılan merkezi tablo (fact table) üretim tablosudur. Boyutlarımız ürün tipi, status(işin aktiflik durumu), üretim aşamaları ve bağlı olan makinalar, son olarak zaman boyutudur (Bkz. Şekil 5.6). Hazırlanan yapı, kar tanesi (Snowflake) şemasına uygun olarak tasarlanmıştır. Üretim küpünün altyapısı hazırlandıktan sonra, standart küp işlemleri SQLServer2000 Enterprise Manager kullanılarak prosedür, fonksiyon ve triger yapısı oluşturularak elde edilmiştir. Kodlama EK2 de verilmiştir. Bu yapı üzerinde yapılan gruplama ve ayrıştırma işlemleri ile istenen çıktılara ulaşılmıştır. SQLServer2000 profiler aracı ile her iki tip

Şekil 5.7 ve Şekil 5.8 de yer alan uygulama arayuzlerinde de görüldüğü gibi uygulamanın çalıştırılabilmesi için; teslim zamanı belirlenecek ürünün seçimi, sipariş miktarının girilmesi, makina üretim kapasitesinin hesaplanması için zaman periyotunun seçilmesi ve hangi yöntem kullanılacak ise seçilmesi gerekmektedir. Stoktaki ürün sayısı çok fazla olduğu için, sıralı tek tek arama mümkün olmamaktadır. Bu nedenle kayıta kolay ulaşılabilmesi için stok adı yazılı alan yazılıma eklenmiştir. Sipariş miktarı, üründen talep edilen miktar içindir. Kapasite hesaplama şekli ise, geçmiş veriler için ne kadar geriye gidileceğini göstermektedir. Bir yılı seçersek son bir yıldaki üretim verileri dikkate alınarak makinanın fireleri dahil gerçek kapasitesi hesaplanacaktır. Ürünlerin kodu, ürünün geçeceği aşamaları ve aşama sırasını belirlemektedir. Ürünler farklı aşama ve sıralardan geçebilmektedir. Bu nedenle her bir ürün için ayrı ayrı hesaplama yapılması gerekmektedir. Uygulama, erken teslim mantığı ile siparişin en erken teslim edilebileceği makinalara atamasını yapmaktadır. Arayüzün alt kısmındaki tabloda aktifler atama yapılan makinaları, pasifler ise atama yapılmayan mevcut makinaları, süreçler bazında göstermektedir. İşlem zamanı ise gerekli seçimler yapıp uygulama çalıştırıldıktan sonra tamamlanması için geçen süreyi göstermektedir. Şekil 5.7 önerdiğimiz yeni mimari ile, Şekil 5.8 ise mevcut VTYS yaklaşımı ile elde edilen sonuçtur. Elde edilen makina ve teslim zamanı ile ilgili sonuçlar aynı iken, gereken işlemci zamanı yaklaşık 400 kat farklı olarak ortaya çıkmıştır. Farklı ürünlerde bu oran 2500'e kadar çıkmıştır.

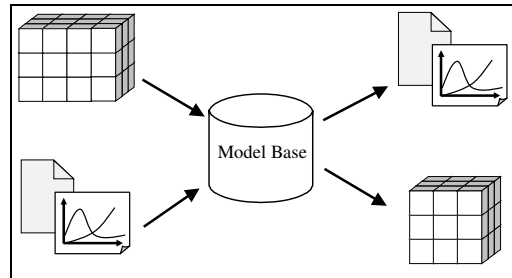
Şekil 5.9'da arayüz çıktılarına bağlı olarak her iki tip yapının da aynı GANT çıktısını verdiği görülmektedir. Şekil 5.7 ve Şekil 5.8'de verilen arayüz çıktılarından elde edilen işlemci zamanı (CPU TIME) her iki tip arasındaki performans farkı ortaya konmuştur. VTYS ile elde edilen işlemci zamanı 1293780 ms. dir. Bu değer 21,5 dakikaya karşılık gelmektedir. Aynı uygulama BVKS yaklaşımı ile yapıldığında elde edilen işlemci zamanı 3330 ms. dir. Yaklaşık 3.3 sn. gibi çok kısa bir zamana indirilmiştir. Çizelge 5.1'de OLAP-VTYS işlem zamanları verilmiştir.



Şekil 5. 9. BVKS çizelgeleme uygulaması ile elde edilen GANT çıktısı

Bu kısımda yapılan uygulama öncelikli olarak performans üstünlüğü ortaya konmuştur. Ayrıca yeni mimari de ortaya konan;(Şekil 5.10.)

- α :Sadece veri küplerinin kullanıldığı yapıdır. Modül içerisinde kodlaması yapılmış analitik algoritmalar (SQL komutları vb.) amaca yönelik olarak yeni küp yapısı
- β :Küplere ek olarak, sisteme harici bir girdi ya da çıktı (analitik rapor, form vb.) dâhil edilmesi ile ortaya çıkan yapıları uygulama ile gerçekleştirilmiştir.



Şekil 5. 10. Önerilen mimaride α , β durumlarının uygulamadaki yapısı

Çizelge 5. 1. Standart VTYS ile Önerilen BVKS için CPU zamanları karşılaştırma tablosu

Ürün Tipi	Sipariş Miktarı	Periyot	Farklı Durumlar İçin Cpu Zamanı (ms)									
			VTYS					BVKS				
4H.....	400	Tüm Veriler	3608390	3699266	3807846	3803576	3642500	1266	1266	1283	1280	1283
		1 Yıl	788543	816750	896576	989860	868313	1250	1250	1250	1263	1250
		3 Aylık	2879860	2752906	2788700	3069390	2968873	1266	1263	1266	1266	1250
		1 Ay	1740343	1808423	1735046	1669203	1668186	1266	1263	1263	1263	1250
		1 Hafta	1334266	1291953	1276203	1292000	1377250	1250	1250	1283	1266	1236
	2500	Tüm Veriler	3603326	3557799	3701542	3799033	3584406	1263	1263	1266	1250	1283
		1 Yıl	896050	857328	882054	806077	784081	1250	1250	1266	1266	1250
		3 Aylık	2762046	2830166	2822876	2872092	2803515	1266	1266	1263	1266	1266
		1 Ay	1663686	1720372	1662440	1712033	1783892	1266	1266	1283	1250	1266
		1 Hafta	1344793	1324584	1271284	1344395	1285640	1266	1280	1266	1263	1266
	10000	Tüm Veriler	3842163	3472239	3656355	3781895	3575837	1266	1250	1266	1263	1266
		1 Yıl	807920	878650	806196	681106	854556	1266	1250	1250	1250	1250
		3 Aylık	2951973	3005874	2897838	2882838	2960844	1296	1266	1266	1266	1263
		1 Ay	1574002	1659713	1738477	1667034	1738077	1266	1313	1266	1250	1266
		1 Hafta	1300603	1283344	1390816	1425331	1257341	1263	1266	1280	1266	1250

Çizelge 5. 2. Standart VTYS ile Önerilen BVKS için CPU zamanları karşılaştırma tablosu

Ürün Tipi	Sipariş Miktarı	Periyot	Farklı Durumlar İçin Cpu Zamanı (ms)									
			VTYS					BVKS				
4T.....	400	Tüm Veriler	3929656	3657733	3627546	3637516	3606966	1440	1483	1436	1453	1483
		1 Yıl	856360	836140	895486	913033	1037390	1500	1453	1440	1466	1453
		3 Aylık	3069483	2663903	2953310	2762200	2812076	1470	1470	1453	1470	1470
		1 Ay	1675593	1674986	1694313	1909313	1775153	1453	1486	1420	1436	1440
		1 Hafta	1415953	1398873	1381033	1315420	1304940	1450	1423	1423	1436	1436
	2500	Tüm Veriler	3831173	3675228	3545426	3747535	3884610	1452	1453	1461	1443	1448
		1 Yıl	915049	893294	895095	871743	919572	1411	1454	1440	1476	1455
		3 Aylık	2831894	2967765	2677642	2755418	3161555	1465	1451	1443	1432	1475
		1 Ay	1975295	1657739	1722878	1691697	1740559	1444	1489	1418	1457	1429
		1 Hafta	1386429	1292008	1330440	1294684	1364304	1419	1437	1432	1467	1438
	10000	Tüm Veriler	3425124	3683999	3860514	3570787	3623441	1422	1480	1431	1460	1469
		1 Yıl	955809	880621	838064	915502	907455	1442	1447	1473	1469	1453
		3 Aylık	2832141	2649435	3103316	2810139	2866578	1420	1456	1467	1460	1463
		1 Ay	1743184	1531878	1748721	1574371	1645800	1480	1442	1477	1448	1495
		1 Hafta	1474204	1347821	1262922	1324755	1449931	1457	1462	1466	1435	1438

Çizelge 5. 3. Standart VTYS ile Önerilen BVKS için CPU zamanları karşılaştırma tablosu

Ürün Tipi	Sipariş Miktarı	Periyot	Farklı Durumlar İçin Cpu Zamanı (ms)									
			VTYS					BVKS				
2A.....	400	Tüm Veriler	2250076	2036170	1936750	2073280	2033720	813	826	830	826	813
		1 Yıl	483296	453436	434373	453436	451813	826	830	830	830	826
		3 Aylık	1539970	1561263	1499016	1527283	1537830	813	843	826	826	846
		1 Ay	917810	915876	927936	1003783	917093	826	830	813	843	810
		1 Hafta	711606	663780	689356	755766	707283	813	826	826	846	843
	2500	Tüm Veriler	2000093	2001346	2094002	2092843	1910330	856	860	860	860	856
		1 Yıl	455436	428629	468744	469871	456060	860	876	860	860	876
		3 Aylık	1607593	1516631	1529638	1517720	1579516	876	856	876	893	846
		1 Ay	929126	925115	888050	945762	977690	843	890	860	843	876
		1 Hafta	758576	726393	654440	731410	747170	856	860	856	890	876
	10000	Tüm Veriler	1988263	1815661	2039151	2191538	1883063	856	860	843	860	860
		1 Yıl	430873	435453	460224	467704	428686	890	843	843	890	890
		3 Aylık	1455970	1581464	1524271	1501787	1437216	860	876	873	860	860
		1 Ay	890733	1002220	853698	927619	882096	876	860	843	856	860
		1 Hafta	667373	715063	729414	701678	675123	860	893	856	860	860

5.3. Uygulama Sonuçlarına Yönelik Deney Tasarımı

Deney tasarımı, 1920’lerde, istatistik biliminin babası sayılan İngiliz istatistikçi Sir Ronald Fisher tarafından, tarım alanında araştırmalar yaparken bulunmuş ve geliştirilmiştir. Fisher ayrıca, deney verilerinin analizi için bugün klasik sayılan “varyans analizi” (ANOVA) yöntemini de geliştirmiştir. Yöntem, tarım alanında çeşitli gübre ve dozları ile iklim koşullarının ve sulama düzeylerinin çeşitli ürünlere olan etkilerini belirlemek üzere uygulanmıştır. [Şirvancı, 1996] Uygulamada kullanılan ürün tipi, veri kullanma periyodu, sipariş miktarı ve yöntemlerden hangilerinin sonuç (CPU işlem zamanı) üzerinde etkili olduğunu görmek için deney tasarımı yapılmıştır. Çizelge 5.1 – Çizelge 5.3 uygulama sonuçlarını kapsayan, SPSS yazılımı kullanılarak yapılan Deney tasarımında Univariate Analysis of Variance ile elde edilen sonuçları Çizelge 5.4 ve takip eden çizelgelerde verilmiştir.

Çizelge 5. 4. Çok değişkenli varyans analizi sonuçları

Değişkenlik Kaynağı	Hata Kareleri Toplamı	Serbestlik Derecesi	Ortalama Kareler	F Değeri	Anlamlılık Değeri
Model	584553511384741,000(a)	89	6568016981851,020	1658,276	,000
Teknik	352917972868682,300	1	352917972868682,300	89103,811	,000
UrunTipi	23406197452029,320	2	11703098726014,660	2954,768	,000
Sipariş Miktarı	14446429509,498	2	7223214754,749	1,824	,163
Periyot	87000134759232,300	4	21750033689808,090	5491,392	,000
Hata	1425870216375,202	360	3960750601,042		
Toplam	939845925559872,000	450			

Modellenen deney tasarımı için kullanılan değişkenler teknik, ürün tip, sipariş miktarı ve periyottur. Çizelge 5.4’de görüldüğü üzere çok değişkenli varyans analizi sonucunda teknik, ürün tipi, sipariş miktarı ve periyodun anlamlılık değeri incelendiğinde kaynaklardan, sipariş miktarının %5 anlamlılık düzeyinde model üzerinde anlamlı etkisi yoktur. Diğer değişkenler modeli etkilemektedir.

Modeli etkileyen değişkenlerin hangi düzeylerinin etkili olduğunu belirlemek amacıyla yapılan Duncan testi sonuçları Çizelge 5.5. - Çizelge 5.7.’de sunulmuştur.

Çizelge 5. 5. Ürün tipi için Duncan Testi sonuçları

UrunTipi	N	Alt Küme	
		1	2
1,00	150	564296,1000	
2,00	150		1042903,4800
3,00	150		1053125,7533
Anlamlılık		1,000	,339

Ürün tipi için iki alt küme oluşmuştur. Birinci kümede ürün tipi 1 vardır. Ürün tipi 2 ve 3 ise ikinci kümeye atılmıştır. Sonuç olarak ürün tipi 1 ile ürün tipi 2 ve 3 arasında anlamlı bir fark olduğu söylenebilir.

Çizelge 5. 6. Sipariş miktarı için Duncan Testi sonuçları

Sipariş Miktarı	N	Alt Küme
		1
10000,00	150	879289,9667
2500,00	150	888040,9467
400,00	150	892994,4200
Anlamlılık		,144

Çizelge 5. 7. Periyot için Duncan Testi sonuçları

Periyot	N	Alt Küme				
		1	2	3	4	5
2,00	90	366402,0000				
5,00	90		564792,6889			
4,00	90			725117,2667		
3,00	90				1212638,9333	
1,00	90					1564924,6667
Anlamlılık		1,000	1,000	1,000	1,000	1,000

Sipariş miktarı için yapılan Duncan testi sonucunda ele alınan her bir düzeyin diğer düzeyle arasında anlamlı bir fark yoktur. Bu nedenle hepsi tek bir alt kümede toplanmıştır. Periyot için yapılan Duncan testi sonuçlarına baktığımızda ise ele alınan her bir düzeyin bir diğeriyle anlamlı bir farklılık oluşturduğu gözlemlenmiştir. Bu nedenle her bir düzey için farklı bir alt küme oluşturulmuştur.

6. SONUÇLAR

Modern kurumlardaki verilerin her geçen sene katlanarak arttığı bilindiğine göre, bu veri yığını ile neler yapılabileceğinin düşünülmesi gerekmektedir. Oluşturulacak bir veri ambarı ile veriler uygun şekilde tutulacak ve OLAP tekniği kullanılarak VM ya da farklı operasyonel amaçlı yaklaşımlar uygulanabilecektir. Bu çalışmada OLAP tabanlı yeni bir mimari olan Bütünleşik Veri Küpü Sistemi önerilmektedir. Sistem merkezinde yönlendirme ve işleme görevi yapan model tabanlı bir modül yer almaktadır. Modül, veri küplerini çok boyutlu dizilere aktaran ve formül kümeleri ile çalışan bir mekanizmadır. Mevcut OLAP işlemleri sınırlı yetenekler sunmaktadır. Yeni önerilen yapı ile, veri küpü, algoritma ve paket programların ilişki içinde olduğu BVKS, OLAP avantajlarını kullanarak, her alanda problem çözümüne imkan sağlayan esnek yapılı bir mimari sunmaktadır. Bilinebildiği kadarıyla literatürde benzeri bir yaklaşım bulunmamaktadır. VA üzerinde OLAP avantajlarını kullanarak analitik teknikler kullanımına imkan sağlayan sistemler süreçlerde iyileşmeye, stratejik kararlar alınmasına taban oluşturacaktır. Bunun anlamı, önerilen yeni yapının tamamının, ciddi bir Karar Destek altyapısı sunmasıdır. Karar Destek Sistemleri, yöneticilerin taktik ve yoğun olarak stratejik kararlarına kolaylıklar sağlayan, ona model tabanındaki modellerle yardımcı olan sistemlerdir. Bu açıdan bakıldığında elde edilen hız ile anında ihtiyacı karşılayan sistemler, rakiplerine avantaj sağlayacaktır.

Çalışmanın satış küpü uygulamasında; OLAP araçlarından (SQL Server) ve .NET 2005-C# kodlama dili kullanılmıştır. Analitik modellere girdi sağlayan küp yapılarından elde edilecek sonuçlar daha sonra yeni küp yapılarına ve sonrasında veri küplerine aktarılmıştır. Böylece, VA'dan elde edilen satış küpünden farklı yeni bir küp elde edilmiştir. BVKS ile ortaya konan durumlardan birisi olan bir küpten yeni bir küpün elde edilmesi durumu gerçekleştirilmiştir. Ayrıca; uygulamaya regresyon katsayılarının bulunması aşamasında, SPSS yazılımı etkileşimli bir şekilde dahil edilmiştir. Elde edilen yeni küpün hesaplama değerleri ile ilgili regresyon raporlarında çıktı olarak alınmıştır. Sonuç olarak satış küpü uygulaması ile bir paket programın ve raporların dahil olduğu, küpten yeni bir küp elde edilmesi durumu ortaya konmuştur.

VTYS-BVKS performans karşılaştırılması uygulamasında, öncelikli olarak sorgu hızlarını ortaya koymanın yanısıra küp üzerinde analitik algoritma çalıştırılarak yeni küplerin ve raporların elde edilebileceğide gösterilmiştir. Mevcut üretim VA'ndan uygulamaya girdi sağlayacak, üretim küpü oluşturulmuştur. Termin zamanını belirlemek amacıyla makina kapasiteleri seçilen zaman aralığı için küp üzerinden hesaplanmıştır. Sonrasında, uygun makinalara erken teslimye yönelik çizelgeleme algoritması kullanarak atamalar yapılmış ve Gant diyagramı çizilmiştir. Bahsedilen süreç VTYS yaklaşımı ve BVKS yaklaşımı için aynı şekilde gerçekleştirilmiş ve benzer sistemlerde en önemli kriter olan sorgu hızı iki sistem için ayrı ayrı test edilmiştir.

Klasik VTYS sorgulamaları ile önerilen BVKS yapısının kullanımı ile edilen uygulamaların performansları sorgulama hızı açısından karşılaştırılmış ve gerçek bir veri kümesi ile Organizasyonel VT üzerinde çizelgeleme uygulaması yapılmıştır. BVKS uygulamasında imalat sektöründeki bir fabrikanın, müşterinin talep ettiği ürün için teslim zamanına çok hızlı cevap verebilen bir sistem ortaya konulmuştur. VTYS ile ortalama sonuç alma süresi 30 dakika sürmektedir. BVKS yaklaşımı ile bu süre 3 sn. ye kadar düşmüştür. Termin zamanı hesaplanacak ürüne bağlı olarak üretim aşamaları değiştiği ve dolayısıyla makina grup ve sayıları değiştiği için sonuçlar farklı ürünlerde değişiklik göstermektedir. Ayrıca, makina kapasitesi hesaplaması için kullanılan geçmiş dönem verileri zaman aralığıda sorgu hızını etkilemektedir. İşlem zamanına hangi faktörlerin etki ettiği deney tasarımı çalışması ile belirlenmiştir. Varyans analizi sonuçlarından da görüldüğü gibi sadece sipariş miktarının sorgu hızı üzerinde anlamlı bir etkisi yoktur. Ürün, zaman aralığı ve kullanılan yöntemin sonuç üzerinde anlamlı bir etkisi vardır. Bunun nedeni bu üç seçeneğin, sorgulama yapılacak hacmi belirlemesidir. Pratikte VTYS ile karşılanması imkansız olan ihtiyaç, VK avantajlarını kullanarak oluşturulan BVKS ile karşılanmıştır. Müşteriye VTYS aksine, kabul edilebilir kısa bir zamanda cevap verebilen BVKS elde edilmiştir. Önerdiğimiz mimari ile sorgu hızlarında ki artış 2500 kata kadar çıkmıştır. Veri miktarının her geçen gün arttığı bilindiğine göre aradaki farkı gösteren bu oran sürekli artış gösterecektir. Avantajlarının yanısıra önerilen mimariyi ve bu alanda yapılan çalışmaların tümünü kapsayan, önemli iki

dezavantaj vardır. Birincisi üzerinde çalışılan konu ile çok büyük hacimli bir veri kümesine sahip olunması şartı, ikincisi ise her yeni uygulamaya özel yoğun emek harcanması gerektiğidir.

OLAP'ın ticari boyut anlamında gelişimi kaçınılmazdır. Sigortacılık, bankacılık ve perakende sektöründe daha fazla kullanılmasına rağmen, üretim sektöründe diğer önemli sektörleri geride bırakarak ilk on arasına girmiştir. Ürün bazında incelendiğinde müşteriler için fiyattan ziyade sorgu hızını da barındıran fonksiyonelliğin en önemli seçim kriteri olduğu görülmüştür. 2006 OLAP raporuna göre tüm yazılımlarda karşılaşılan en ciddi problem ve şikayet sorgu hızı yavaşlığıdır. Tüm yazılımlarda bu sorunu aşmak için bir çözüm yolu, firmaların uygulamalarında VK yaklaşımını kullanmasıdır.

Endüstride üretim ve karar etkinliğini artıran, literatürde kabul görmüş teknik ve yaklaşımlar vardır. Veri tabanı kullanımından veri ambarı kullanımına geçişin yaygınlaştığı endüstride, inanıyoruz ki bu tezde önerilen yeni mimarinin uygulama konmasıyla ciddi avantajlar sağlanabilecektir. Örneğin; bu yeni mimari, piyasada bulunan tüm ticari MRP II/ERP yazılımlarına adapta edilebilir. VT üzerinde kurulu ERP sistemleri için bu mimarinin kullanımı mevcut sisteme göre üstünlükleri açısından rakiplerine avantaj sağlayacaktır. Yaklaşımda, bileşenler kullanılarak oluşturulan küp sistemi, analitik algoritmalarla desteklenen bir modül ile daha iyi sonuçlar veren bir yapıyı sunmaktadır. Özetle, yeni yapının adaptasyonu sonucunda, bu ticari yazılımlar ile hem çok hızlı sorgu performansı yakalamak, hem de yöneticiye çok seçenekli esnek raporlama imkanları sunmak mümkün olabilecektir. Bu tezle ortaya konulan ve küçük bir uygulama üzerinde test edilen mimarinin, tezin tamamlanmasından ve tezle ilgili yapılan çalışmaların farkedilmesinden sonra ticari yazılım firmaları tarafından da farkedileceği, bilişim teknolojilerindeki hızlı gelişmeler dikkate alındığında, yakın bir zaman diliminde bu mimarinin, ticari firmaların yazılımlarına entegre edileceği (gömüleceği) beklenmekte ve ümit edilmektedir. Ayrıca, bu mimarinin, OLAP konusunda çalışanlara iyi bir referans olacağı da düşünülmektedir.

KAYNAKLAR

Adriaans, P., Zantinge, D., “Data Mining”, *Addison Wesley Longman*, England, 37-78 (1996).

Alhajj, R., Kaya, M., “Integrating fuzziness into OLAP for multidimensional fuzzy association rules mining”, *Data Mining, ICDM 2003. Third IEEE International Conference*, Melbourne, Florida, USA, 469- 472 (2003).

Alkharouf, N.W., Jamison, D.C., Matthews, B.F., “Online Analytical Processing (OLAP): A Fast and Effective Data Mining Tool for Gene Expression Databases”, *Journal of Biomedicine and Biotechnology*, 2:181-188 (2005).

Ansari, S., Kohavi, R., Mason, L., Zheng, Z., “Integrating E-Commerce and Data Mining: Architecture and Challenges”, *Proceedings of the 2001 IEEE International Conference on Data Mining*, San Jose, CA, USA, 27-34 (2001).

Berry, M., Linoff, G., “Data Mining Techniques for Marketing Sales and Customer Support”, *John Wiley & Sons*, 2-12 (1997).

Chau, K.W., Cao, Y., Anson, M., Zhang, J., “Application of data warehouse and Decision Support System in construction management”, *Automation in Construction*, 12:213-224 (2002).

Chaudhuri, S., Dayal, U., “An Overview of Data Warehousing and OLAP Technology”, *ACM SIGMOD Record*, 26(1): 65-74 (1997).

Chaudhuri, S., Dayal, U., “Data warehousing and OLAP for decision support”, *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, Tucson, Arizona, USA, 507–508 (1997).

Chen, Q., Dayal, U., Hsu, M., “OLAP-based Scalable Profiling of Customer Behavior”, *Proc. Of 1 st International Conference on Data Warehousing and Knowledge Discovery (DAWAK99)*, Italy, 55-64 (1999).

Chen, Z., “Data Mining and Uncertain Reasoning”, *A Wiley-Interscience Publication*, USA, 78-79, (2001).

Codd, E.F., Codd, S.B., Salley, C.T., “Providing OLAP (on-line analytical processing) to user analysts: an IT mandate”, *Technical report, E.F. Codd and Associates* ,1-7 (1993).

Çetinyokuş, T., Gökçen, H., “Bütünleşik Veri Küpü Sistemi (BVKS): Satış Küpü Uygulaması”, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 23(2) 468-477 (2008).

Datta, A., Thomas, H., “The Cube Data Model: a conceptual model and algebra for on-line analytical processing in data warehouses”, *Decision Support Systems*, 27:289-301 (1999).

Dhar, V., Stein, R., “Intelligent Decision Support Methods: The Science of Knowledge Work”, *Prentice Hall*, New Jersey (1997).

Internet: Dilly, R., “Data Mining: An Introduction.” www.pcc.qub.ac.uk/tec/courses/datamining/stu_notes/dm_book_1.html (2005).

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., “From Data Mining to Knowledge Discovery in Databases”, *American Association for Artificial Intelligence*, 37-54 (1996).

Giudici, P., “Applied Data Mining – Statistical Methods for Business and Industry”, *John Wiley & Sons*, Sussex, 1-14 (2003).

Glymour, C., Madigan, D., Pregibon, D., Smyth, P., “Statistical themes and lessons for data mining”, *Data Mining and Knowledge Discovery*, 25-42 (1996).

Goebel, M., Gruenwald, L., “A Survey of Data Mining and Knowledge Discovery Software Tools”, *SIGKDD Explorations*, 1(1): 20-33 (1999).

Goil, S., Choudhary, A., “High Performance OLAP and Data Mining on Parallel Computers”, *Data Mining and Knowledge Discovery*, 1:391-417 (1997).

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H., “Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals”, *J. Data Mining and Knowledge Discovery*, 1:29-53 (1997).

Gupta, H., Harinarayan, V., Rajaraman, A., Ullman, J. D., “Index selection for OLAP”, *In Proc. 13th ICDE*, Manchester, UK, 208-219 (1997).

Han, J., “OLAP mining: An integration of OLAP with data mining”, *In Proceedings of the 7th IFIP 2.6 Working Conference on Database Semantics (DS-7)*, Leysin, Switzerland, 1-9 (1997).

Han, J., Kamber, M., “Data Mining Concept and Techniques”, *Morgan Kaufman*, San Francisco, 39-99 (2001).

Hand, D.J., “Statistics and data mining: intersection discipline”, *Knowledge Discovery & Data Mining, ACM*, 16-19 (1999).

Hu, X., Cercone, N., “A data warehouse/online analytic processing framework for web usage mining and business intelligence reporting”, *International Journal of Intelligent Systems*, 19:585-606 (2004).

Kaya, M., Alhajj, R., “Fuzzy OLAP Association Rules Mining-Based Modular Reinforcement Learning Approach for Multiagent Systems”, *IEEE Transactions on Systems, Man, and Cybernetics*, 35:2 (2005).

Kovalerchuk, B., “Data Mining in Finance: Advances in Relational and Hybrid Methods”, *Kluwer Academic*, New York, 1-19 (2000).

Lau H.C.W., Jiang, B., Lee, W.B., Lau, K.H., “Development of an intelligent data-mining system for a dispersed manufacturing network”, *Expert Systems*, 18(4):175-185 (2001).

Laudon, C. K., Laudon, J. P., “Management Information Systems, 7th Edition”, *Prentice Hall*, New Jersey, 10-12, (2002).

Leung, R.W.K., Lau, H.C.W., Kwong, C.K., “An expert system to support the optimization of ion plating process: an OLAP-based fuzzy-cum-GA approach”, *Expert Systems with Applications*, 25:313-330 (2003).

Lin, W.Y., Kuo, I.C., “OLAP data cubes configuration with genetic algorithms”, *Systems, Man, and Cybernetics, IEEE International Conference*, Nashville, TN, 3:1984-1989 (2000).

Lin, W.Y., Kuo, I.C., “A Genetic Selection Algorithm for OLAP Data Cubes”, *Knowledge and Information Systems*, 6:83-102 (2004).

Maniatis, A., Vassiliadis, P., Skiadopoulos, S., Vassiliou, Y., “CPM: A Cube Presentation Model for OLAP”, *Data Warehousing and Knowledge Discovery, 5th. International Conference*, DaWak, 2003:4-13 (2003).

Marakas G. M., “Decision Support Systems”, *Prentice Hall*, New Jersey, 21-57 (1999).

Maxon, B., “Defining Data Mining”, *DBMS Data Warehouse Supplement*, 9(9): 7-13 (1996).

Palpanas, T., “Knowledge Discovery in Data Warehouses”, *Sigmod Record*, 29(3):88-100 (2000).

Pendse, N., “The OLAP Survey 6”, *BOARD Survey*, 1-23 (2006).

Piatetsky-Shapiro, G., Frawley, W., J., “Knowledge Discovery in Database”, *AAAI MIT Press*, 4-7 (1991).

Raisinghani, M., “Business Intelligence in the Digital Economy: Opportunities, Limitations and Risks”, *Hershey*, PA, USA, 142-145 (2004).

Sang, J. L., Keng, S., “A review of data mining techniques”, *Industrial Management & Data Systems*, 25-36 (2001).

İnternet: SAS Institute Inc., “The Data Mining Challenge: Turning Raw Data Into Business Gold”, www.sas.com/software/data_mining/ (2005).

Shi, Z., Huang, Y., He, Q., Xu, L., Liu, S., Qin, L., Jia, Z., Li, J., Huang, H., Zhao, L., “MSMiner-A Developing Platform for OLAP”, *Decision Support Systems*, 42(4):2016-2028 (2007).

İnternet: SPSS Inc., “AnswerTree 2.0 User’s Guide”, ISBN 1-56827-254-5, www.spss.com/be/distr/dutch/bookshop/index.html (1998).

İnternet: SPSS Inc., “More on What Data Mining is – and isn’t”, www.spss.com/datamine/what2.htm (2005).

Şirvancı, M., “Kalite için Deney Tasarımı 'Taguçi Yaklaşımı'”, *Literatür Yayıncılık*, Milwaukee, ABD, 3-4 (1996).

Thalhammer, T., Schrefl, M., Mohania, M., “Active data warehouses: complementing OLAP with analysis rules”, *Data & Knowledge Engineering*, 39:241-269 (2001).

Theodoratos, D., Sellis, T., “Answering Multidimensional Queries on Cubes Using Other Cubes”, *Scientific and Statistical Database Management, Proceedings. 12th International Conference*, Berlin, Germany, 109-122 (2000).

Tseng, F.S.C., “Design of a multi-dimensional query expression for document warehouses”, *Information Sciences*, 174:55-79 (2005).

Vaisman, A.A., Mendelzon, A.O., Ruaro, W., Cymerman, S.G., “Supporting dimension updates in an OLAP server”, *Information Systems*, 29:165-185 (2004).

Vassiliadis, P., “Modeling multidimensional databases, cubes and cube operations”, *Proc. 10 th Int. Conference on Statistical and Scientific Database Management (SSDBM)*, Capri, Italy, 53-62 (1998).

Waterhouse, P., “1998 Entertainment media and communications technology forecast”, *Price Waterhouse World Technology Center*, Menlo Park, CA, 3 (1997).

Weiss, S.H., Indurkha, N., “Predictive Data Mining: A practical Guide”, *Morgan Kaufmann*, San Francisco, 11-14 (1998).

Westphal, C., Blaxton, T., “Data Mining Solutions”, *Wiley Computer Publishing*, USA, 201-240 (1998).

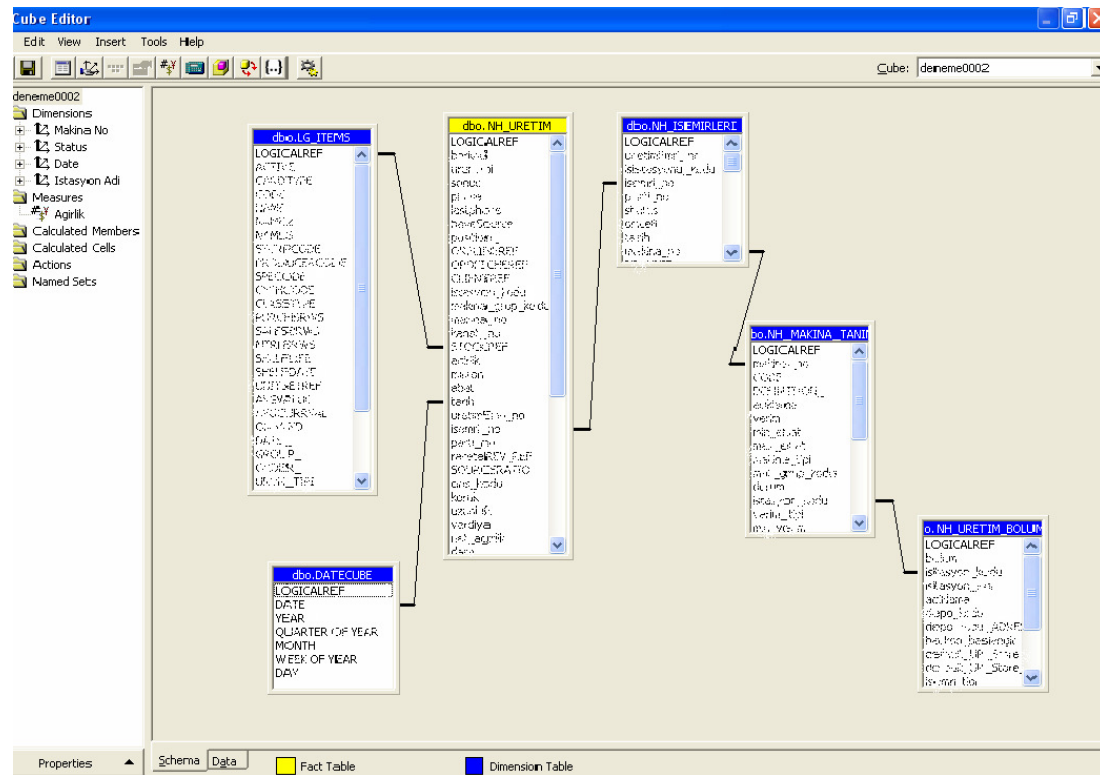
Witkowski, A., Bellamkonda, S., Bozkaya, T., Folkert, N., Gupta, A., Haydu, J., Sheng, L., Subramanian, S., “Advanced SQL modeling in RDBMS”, *ACM Transactions on Database Systems*, 30(1):83-121 (2005).

Zhong, N., Zhou, L., “Methodologies for Knowledge Discovery and Data Mining”, *Springer*, Germany, 23-71 (1999).

Zytkow J.M., Rauch, J., “Methodologies for Knowledge Discovery and Data Mining”, *Springer*, Germany, 21-169 (1999).

EKLER

EK-1 BVKS odaklı OLAP ve VTYS Performans Uygulaması Arayuzleri



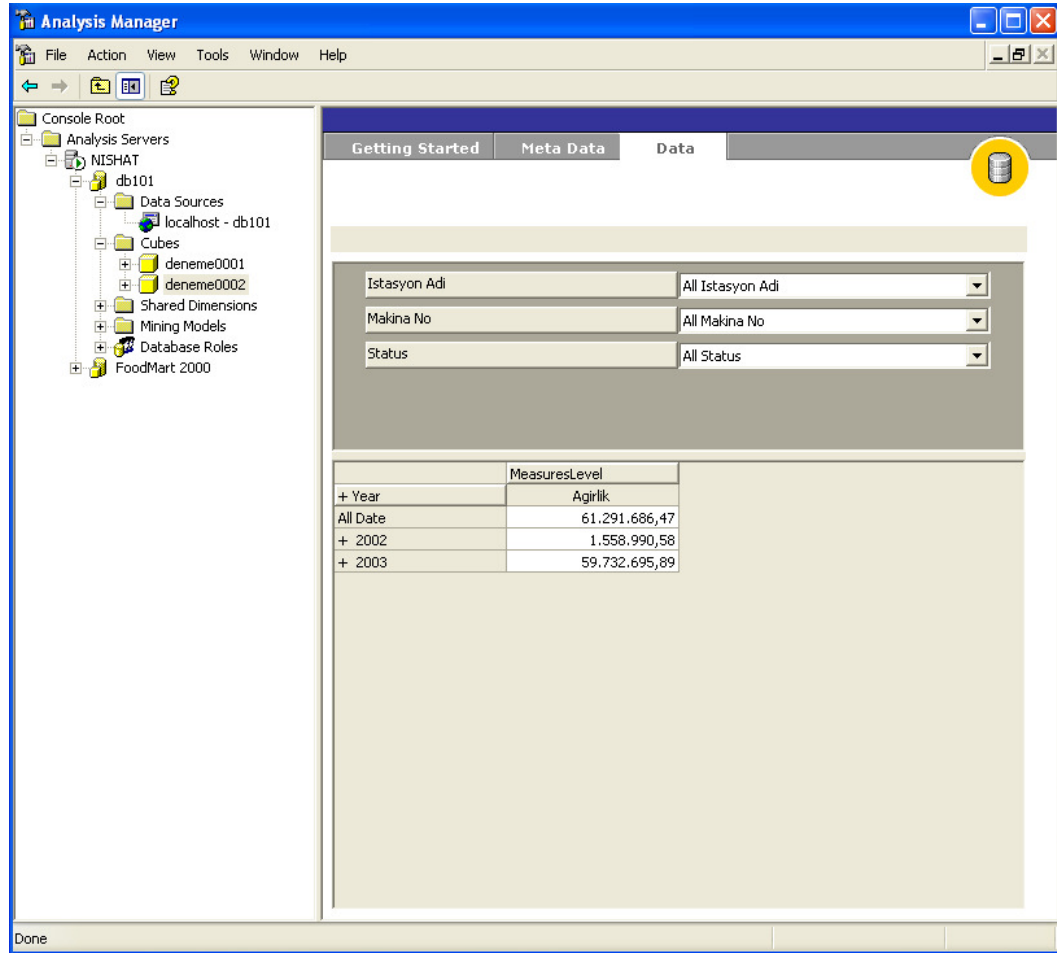
EK-1 (Devam) BVKS odaklı OLAP ve VTYS Performans Uygulaması Arayüzleri

The screenshot displays the Microsoft Analysis Manager interface. On the left, the 'Console Root' tree shows the hierarchy: Analysis Servers > NISHAT > db101 > Cubes > deneme0002. The main pane shows the 'Data' tab for the selected cube.

Cube: deneme0002

Property	Value
Dimensions:	Makina No, Status, Date, Istasyon Adi
Measures:	Agirlik
Calculated Members:	None
Calculated Cells:	None
Actions:	None
Named Sets:	None
Source Tables:	dbo.NH_URETIM, dbo.NH_ISEMIRLERI, dbo.NH_MAKINA_TANIM, dbo.NH_URETIM_BOLUMU, dbo.DATECUBE
Fact Table:	dbo.NH_URETIM

EK-1 (Devam) BVKS odaklı OLAP ve VTYS Performans Uygulaması Arayüzleri



EK-1 (Devam) BVKS odaklı OLAP ve VTYS Performans Uygulaması Arayüzleri

Cube Browser - deneme0002

Istasyon Adı: All Istasyon Adı
Status: All Status

				MeasuresLevel	Makina No			
				Ağırlık				
- Year	- Quarter	- Month	Day	All Makina No	100	201	202	
All Date	All Date Total			61.291.686,47	761,44	1.546.576,04		3
	2002 Total			1.558.990,58		41.360,10		
		Quarter 3 Total		160.277,25				
			August Total	51.944,77				
			20	5.303,84				
			21	4.829,28				
			22	2.708,32				
			23	1.007,76				
			24	2.577,85				
			26	500,80				
			27	3.694,74				
			28	4.825,30				
			29	6.911,66				
			30	8.862,66				
			31	10.722,56				
			September Total	108.332,48				
		+ September						
	+ Quarter 4	Quarter 4 Total		1.398.713,33		41.360,10		
+ 2003	2003 Total			59.732.695,89	761,44	1.505.215,94		3

Double-click a member to drill up or down.

Close Help

EK-1 (Devam) BVKS odaklı OLAP ve VTYS Performans Uygulaması Arayüzleri

Cube Browser - deneme0002

Istasyon Adi: KESIM
 Status: 0
 Makina No: All Makina No

- Year	- Quarter	- Month	Day	MeasuresLevel
All Date	All Date Total			Ağırlık
+ 2002	2002 Total			2.156.375,39
	2003 Total			441,60
		Quarter 1 Total		2.155.933,79
			January Total	1.279.078,33
			1	389.250,19
			2	13.499,82
			3	14.662,26
			4	9.390,70
			5	14.242,04
			6	12.274,76
			7	19.428,84
			8	15.816,72
			9	13.381,18
			10	7.835,64
			11	13.619,48
			12	18.987,50
			13	17.278,76
			14	13.038,76
			15	11.141,10
			16	13.054,36
			17	9.678,39
			18	11.867,02
			19	7.088,26
			20	7.213,82
			21	14.365,86
			22	18.224,46
			23	12.268,78
			24	12.897,66
			25	14.903,80
				11.881,74

Double-click a member to drill up or down.

Close Help

EK-2 BVKS odaklı OLAP ve VTYS Performans Uygulama Program Kodu

C++ Arayuz Kodu

```
//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Edit1Change(TObject *Sender)
{
    tblStok->Active=false;
    tblStok->Parameters->ParamByName("CODE")->Value=Edit1->Text+'%';
    tblStok->Active=true;
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    anmProcess->Visible=true;
    clearRapor();
    procVTYS->Active=false;
    procOLAP->Active=false;
    procOLAP->Parameters->ParamByName("HESAPTIPI")->Value=hesapTipi->ItemIndex;
    procOLAP->Parameters->ParamByName("STOCKREF")->Value=tblStok-
>FieldByName("LOGICALREF")->AsInteger;
    procOLAP->Parameters->ParamByName("SIPARISMIKTAR")-
>Value=StrToFloat(siparisMiktar->Text);
```

```

procOLAP->Active=true;
teslimZamani->Text=FormatFloat("#,0.00",procOLAP->FieldByName("TESLIM")->AsFloat);
CPU->Text=procOLAP->FieldByName("CPU")->AsString;
showRapor("T_GANT_OLAP");
anmProcess->Visible=false;
}
//-----
void TForm1::showRapor(String f_tabloAdi){
    tblRapor->Active=false;
    tblRapor->CommandText="SELECT * FROM "+f_tabloAdi+" TBL LEFT OUTER JOIN
NH_VIEW_MAKINA_TANIM_1 MT ON MT.MACHINE_LOGICALREF=TBL.MAKINAREF ";
    if(aktifGoster->Checked){
        tblRapor->CommandText=tblRapor->CommandText+" WHERE AKTIF=1";
    }
    tblRapor->Active=true;
}
void __fastcall TForm1::aktifGosterClick(TObject *Sender)
{
    if(procVTYS->Active)
        showRapor("T_GANT_VTYS");
    else
        showRapor("T_GANT_OLAP");
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    anmProcess->Visible=true;
    clearRapor();
    procOLAP->Active=false;
    procVTYS->Active=false;
    procVTYS->Parameters->ParamByName("HESAPTIPI")->Value=hesapTipi->ItemIndex;
    procVTYS->Parameters->ParamByName("STOCKREF")->Value=tblStok-
>FieldByName("LOGICALREF")->AsInteger;
    procVTYS->Parameters->ParamByName("SIPARISMIKTAR")-
>Value=StrToFloat(siparisMiktar->Text);
    procVTYS->Active=true;
    teslimZamani->Text=FormatFloat("#,0.00",procVTYS->FieldByName("TESLIM")->AsFloat);
    CPU->Text=procVTYS->FieldByName("CPU")->AsString;
}

```

```

        showRapor("T_GANT_VTYS");
        anmProcess->Visible=false;
    }
//-----
void TForm1::clearRapor(){
    teslimZamani->Text="";
    CPU->Text="";
    tblRapor->Active=false;
    Form1->Refresh();
}
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    int i,j;

    j=tblRapor->RecordCount;
    RaporSC.resize(j);
    RaporMC.resize(j);
    RaporYS.resize(j);
    RaporYT.resize(j);
    RaporYF.resize(j);
    RaporEF.resize(j);

    tblRapor->First();//ilk kayda gider
    for(i=0;i<tblRapor->RecordCount;i++){
        RaporSC[i]=(AnsiString(tblRapor->FieldByName("STATION_CODE")->AsString));
        RaporMC[i]=(AnsiString(tblRapor->FieldByName("MACHINE_CODE")->AsString));
        RaporYS[i]=(int)(tblRapor->FieldByName("YENIBASLAMA")->AsFloat);
        RaporYT[i]=(int)(tblRapor->FieldByName("YENISURE")->AsFloat);
        RaporYF[i]=(int)(tblRapor->FieldByName("YENIBITIS")->AsFloat);
        RaporEF[i]=(int)(tblRapor->FieldByName("MEVCUTBITIS")->AsFloat);
        //ShowMessage(AnsiString(RaporSC[i]));
        tblRapor->Next();
    }
    //this->Hide();
    FrmGanth->Active;
    //FrmGanth->DrawGanth(RaporYS, RaporYF, RaporEF, RaporSC, RaporMC);
}

```

```

        FrmGanth->ShowModal();
    }
    // ShowMessage(FloatToStr(tblRapor->FieldByName("KAPASITE")->AsFloat));

```

SQLServer Veri K p  Kodu

```

CREATE PROCEDURE T_EXEC_TOPLAM_MAKINA_ZAMAN AS
DROP TABLE T_TOPLAM_MAKINA_ZAMAN
SELECT
        MT.STATIONCODE,
        MT.MACHINEREF,
        datepart(year,DATE_) AS YIL,
        datepart(quarter,DATE_) AS CEYREKYIL,
        datepart(month,DATE_) AS AY,
        datepart(week,DATE_) as HAFTA,
        SUM(WEIGHT) AS TURETIM,
        MIN(DATE_) AS BASLAMA,
        MAX(DATE_) AS BITIS
        INTO T_TOPLAM_MAKINA_ZAMAN
FROM
        TBL_PRODUCTION U LEFT OUTER JOIN
        TBL_MACHINEDEFINATIONS MT ON MT.MACHINEREF=U.MACHINEREF
GROUP BY
        MT.STATIONCODE,
        MT.MACHINEREF,
        datepart(year,DATE_),
        datepart(quarter,DATE_),
        datepart(month,DATE_),
        datepart(week,DATE_)
ORDER BY
        MT.STATIONCODE,
        MT.MACHINEREF,
        datepart(year,DATE_),
        datepart(quarter,DATE_),
        datepart(month,DATE_),
        datepart(week,DATE_)

SELECT 'ISLEM TAMAM' AS SONUC
GO

```

```

CREATE FUNCTION T_KAPASITE_OLAP (@HESAPTIPI INT,@ASAMAREF
INT,@MAKINAREF VARCHAR(10))
RETURNS FLOAT AS
BEGIN
/* HESAP TIPI İÇİN ----->0:HAFTA,1:AY,2:CEYREKYIL,3:YIL,4:ALL*/
DECLARE
    @TARİH DATETIME,
    @KAPASITETARİHI DATETIME,
    @KAPASITEYIL VARCHAR(10),
    @KAPASITECEYREKYIL VARCHAR(10),
    @KAPASITEAY VARCHAR(10),
    @KAPASITEHAFTA VARCHAR(10),
    @URETIMMIKTARI FLOAT,
    @URETIMSAATI INT,
    @KAPASITE FLOAT

SET @TARİH=(SELECT MAX(DATE_) FROM TBL_PRODUCTION)
SET @KAPASITETARİHI=(CASE(@HESAPTIPI)
    WHEN 0 THEN DATEADD(Week,-1,@TARİH)
    WHEN 1 THEN DATEADD(Month,-1,@TARİH)
    WHEN 2 THEN DATEADD(Quarter,-1,@TARİH)
    WHEN 3 THEN DATEADD(Year,-1,@TARİH)
END)
IF(@HESAPTIPI<=3) SET @KAPASITEYIL=DATEPART(Year,@KAPASITETARİHI) ELSE
SET @KAPASITEYIL='%'
IF(@HESAPTIPI<=2) SET
@KAPASITECEYREKYIL=DATEPART(Quarter,@KAPASITETARİHI) ELSE SET
@KAPASITECEYREKYIL='%'
IF(@HESAPTIPI<=1) SET @KAPASITEAY=DATEPART(Month,@KAPASITETARİHI) ELSE
SET @KAPASITEAY='%'
IF(@HESAPTIPI<=0) SET @KAPASITEHAFTA=DATEPART(Week,@KAPASITETARİHI)
ELSE SET @KAPASITEHAFTA='%'

SELECT
    @URETIMMIKTARI=ISNULL(SUM(TURETIM),0),
    @URETIMSAATI=ISNULL(DateDiff(Hour,MIN(BASLAMA), MAX(BITIS)),0)
FROM
    T_TOPLAM_MAKINA_ZAMAN
WHERE
    STATIONREF=@ASAMAREF AND
    MACHINEREF=@MAKINAREF AND
    YIL LIKE @KAPASITEYIL AND
    CEYREKYIL LIKE @KAPASITECEYREKYIL AND
    AY LIKE @KAPASITEAY AND
    HAFTA LIKE @KAPASITEHAFTA
SET @KAPASITE=@URETIMMIKTARI/(CASE(@URETIMSAATI) WHEN 0 THEN 1 ELSE
@URETIMSAATI END)
RETURN @KAPASITE
END

CREATE FUNCTION T_KAPASITE_VTYS (@HESAPTIPI INT,@ASAMAREF
INT,@MAKINAREF VARCHAR(10))
RETURNS FLOAT AS
BEGIN
/* HESAP TIPI İÇİN ----->0:HAFTA,1:AY,2:CEYREKYIL,3:YIL,4:ALL*/
DECLARE

```

```

    @TARIH DATETIME,
    @KAPASITETARIHI DATETIME,
    @KAPASITEYIL VARCHAR(10),
    @KAPASITECEYREKYIL VARCHAR(10),
    @KAPASITEAY VARCHAR(10),
    @KAPASITEHAFTA VARCHAR(10),
    @URETIMMIKTARI FLOAT,
    @URETIMSAATI INT,
    @KAPASITE FLOAT

SET @TARIH=(SELECT MAX(DATE_) FROM TBL_PRODUCTION)
SET @KAPASITETARIHI=(CASE( @HESAPTIPI)
    WHEN 0 THEN DATEADD(Week,-1,@TARIH)
    WHEN 1 THEN DATEADD(Month,-1,@TARIH)
    WHEN 2 THEN DATEADD(Quarter,-1,@TARIH)
    WHEN 3 THEN DATEADD(Year,-1,@TARIH)
END)
IF(@HESAPTIPI<=3) SET @KAPASITEYIL=DATEPART(Year,@KAPASITETARIHI) ELSE
SET @KAPASITEYIL='%'
IF(@HESAPTIPI<=2) SET
@KAPASITECEYREKYIL=DATEPART(Quarter,@KAPASITETARIHI) ELSE SET
@KAPASITECEYREKYIL='%'
IF(@HESAPTIPI<=1) SET @KAPASITEAY=DATEPART(Month,@KAPASITETARIHI) ELSE
SET @KAPASITEAY='%'
IF(@HESAPTIPI<=0) SET @KAPASITEHAFTA=DATEPART(Week,@KAPASITETARIHI)
ELSE SET @KAPASITEHAFTA='%'

SELECT
    @URETIMMIKTARI=ISNULL(SUM(U.WEIGHT),0),
    @URETIMSAATI=ISNULL(DateDiff(Hour,MIN(U.DATE_), MAX(U.DATE_)),0)

FROM
    TBL_PRODUCTION U LEFT OUTER JOIN
    TBL_MACHINEDEFINATIONS MT ON MT.MACHINEREF=U.MACHINEREF
WHERE
    MT.STATIONREF=@ASAMAREF AND
    MT.MACHINEREF=@MAKINAREF AND
    DatePart(Year,U.DATE_) LIKE @KAPASITEYIL AND
    DatePart(Quarter,U.DATE_) LIKE @KAPASITECEYREKYIL AND
    DatePart(Month,U.DATE_) LIKE @KAPASITEAY AND
    DatePart(Week,U.DATE_) LIKE @KAPASITEHAFTA
SET @KAPASITE=@URETIMMIKTARI/(CASE( @URETIMSAATI) WHEN 0 THEN 1 ELSE
@URETIMSAATI END)
RETURN @KAPASITE
END

CREATE FUNCTION T_STOKASAMALARI (@STOCKREF INT,@SIPARISMIKTAR
FLOAT)
RETURNS
@ASAMALAR TABLE (
    ASAMAREF INT,
    ASAMAADI VARCHAR(50),
    ASAMASIRA INT,
    ASAMAMIKTAR FLOAT)
AS
BEGIN

```



```

DECLARE @ASAMAREF INT,
        @ASAMAADI VARCHAR(50),
        @ASAMASIRA INT,
        @ASAMAMIKTAR FLOAT,
        @SCRAPRATE FLOAT,
        @ASAMACOUNTER INT
SET @ASAMAMIKTAR=@SIPARISMIKTAR
DECLARE CURSOR_ASAMA CURSOR FOR
    SELECT
        UB.STATIONREF,
        UB.STATIONCODE,
        UB.PRODORDER,
        UB.SCRAPRATE
    FROM
        dbo.TBL_PRODUCTIONORDERS UE WITH (NOLOCK) LEFT OUTER
JOIN
        dbo.TBL_PRODUCTIONDEPARTMENTS UB WITH (NOLOCK) ON
UE.STATIONREF = UB.STATIONREF LEFT OUTER JOIN
        dbo.TBL_ITEMDEFINITIONS IT WITH(NOLOCK) ON
IT.LOGICALREF=UE.STOCKREF
    WHERE
        IT.CODE LIKE (SELECT LEFT(IT.CODE,5)+'%' FROM
        dbo.TBL_ITEMDEFINITIONS IT WHERE IT.LOGICALREF=@STOCKREF) AND
        UB.STATIONREF IN (1,2,3,15)
    GROUP BY
        UB.STATIONREF,UB.STATIONCODE,UB.PRODORDER,UB.SCRAPRATE
    ORDER BY
        UB.PRODORDER DESC
OPEN CURSOR_ASAMA

FETCH NEXT FROM CURSOR_ASAMA
INTO @ASAMAREF,@ASAMAADI,@ASAMASIRA,@FIREORANI

WHILE @@FETCH_STATUS=0
BEGIN
    INSERT @ASAMALAR
    SELECT @ASAMAREF,@ASAMAADI,@ASAMASIRA,@ASAMAMIKTAR

    SET @ASAMAMIKTAR=@ASAMAMIKTAR+@ASAMAMIKTAR*@FIREORANI

    FETCH NEXT FROM CURSOR_ASAMA
    INTO @ASAMAREF,@ASAMAADI,@ASAMASIRA,@FIREORANI
END
CLOSE CURSOR_ASAMA
DEALLOCATE CURSOR_ASAMA

RETURN
END

```

```

CREATE PROCEDURE T_SIPARIS_BITIS_FABRIKA_OLAP
    @HESAPTIPI INT,
    @STOCKREF INT,
    @SIPARISMIKTAR FLOAT
AS
DECLARE @ASAMAREF INT,
    @ASAMASIRA INT,
    @ASAMAMIKTAR FLOAT,
    @ASAMASIPARISHAZIROLMA FLOAT,
    @ASAMASLEMSIRA INT,
    @SIPARISBITISASAMA FLOAT,
    @SECILENMAKINAREF VARCHAR(12),
    @ISLEMBASLAMA DATETIME,
    @ISLEMBITIS DATETIME
SET @ISLEMBASLAMA=GETDATE()
SET @ASAMASIPARISHAZIROLMA=0.0
SET @ASAMASLEMSIRA=1

DELETE T_GANT_OLAP

DECLARE      CURSOR_ASAMA CURSOR FOR
SELECT ASAMAREF,ASAMASIRA,ASAMAMIKTAR FROM
    DBO.T_STOKASAMALARI(@STOCKREF,@SIPARISMIKTAR) ORDER BY ASAMASIRA

OPEN CURSOR_ASAMA

FETCH NEXT FROM CURSOR_ASAMA
INTO @ASAMAREF,@ASAMASIRA,@ASAMAMIKTAR

WHILE @@FETCH_STATUS=0
BEGIN

    EXEC T_SIPARIS_BITIS_ASAMA_OLAP
        @HESAPTIPI,
        @ASAMASLEMSIRA,
        @ASAMAREF,
        @ASAMAMIKTAR,
        @ASAMASIPARISHAZIROLMA,
        @SIPARISBITISASAMA OUTPUT,
        @SECILENMAKINAREF OUTPUT

    SET @ASAMASLEMSIRA=@ASAMASLEMSIRA+1
    SET @ASAMASIPARISHAZIROLMA=@SIPARISBITISASAMA

    FETCH NEXT FROM CURSOR_ASAMA
    INTO @ASAMAREF,@ASAMASIRA,@ASAMAMIKTAR

END
CLOSE CURSOR_ASAMA
DEALLOCATE CURSOR_ASAMA

SET @ISLEMBITIS=GETDATE()
SELECT @ASAMASIPARISHAZIROLMA AS
    TESLIM,DATEDIFF(millisecond,@ISLEMBASLAMA,@ISLEMBITIS) AS CPU
GO

```

```

CREATE PROCEDURE T_SIPARIS_BITIS_FABRIKA_VTYS
    @HESAPTIPI INT,
    @STOCKREF INT,
    @SIPARISMIKTAR FLOAT
AS
DECLARE @ASAMAREF INT,
    @ASAMASIRA INT,
    @ASAMAMIKTAR FLOAT,
    @ASAMASIPARISHAZIROLMA FLOAT,
    @ASAMASLEMSIRA INT,
    @SIPARISBITISASAMA FLOAT,
    @SECILENMAKINAREF VARCHAR(12),
    @ISLEMBASLAMA DATETIME,
    @ISLEMBITIS DATETIME
SET @ISLEMBASLAMA=GETDATE()
SET @ASAMASIPARISHAZIROLMA=0.0
SET @ASAMASLEMSIRA=1

DELETE T_GANT_VTYS

DECLARE      CURSOR_ASAMA CURSOR FOR
SELECT ASAMAREF,ASAMASIRA,ASAMAMIKTAR FROM
DBO.T_STOKASAMALARI(@STOCKREF,@SIPARISMIKTAR) ORDER BY ASAMASIRA

OPEN CURSOR_ASAMA

FETCH NEXT FROM CURSOR_ASAMA
INTO @ASAMAREF,@ASAMASIRA,@ASAMAMIKTAR

WHILE @@FETCH_STATUS=0
BEGIN

    EXEC T_SIPARIS_BITIS_ASAMA_VTYS
        @HESAPTIPI,
        @ASAMASLEMSIRA,
        @ASAMAREF,
        @ASAMAMIKTAR,
        @ASAMASIPARISHAZIROLMA,
        @SIPARISBITISASAMA OUTPUT,
        @SECILENMAKINAREF OUTPUT

    SET @ASAMASLEMSIRA=@ASAMASLEMSIRA+1
    SET @ASAMASIPARISHAZIROLMA=@SIPARISBITISASAMA

    FETCH NEXT FROM CURSOR_ASAMA
    INTO @ASAMAREF,@ASAMASIRA,@ASAMAMIKTAR

END
CLOSE CURSOR_ASAMA
DEALLOCATE CURSOR_ASAMA

SET @ISLEMBITIS=GETDATE()
SELECT @ASAMASIPARISHAZIROLMA AS
TESLIM,DATEDIFF(millisecond,@ISLEMBASLAMA,@ISLEMBITIS) AS CPU
GO

```

CREATE PROCEDURE T_SIPARIS_BITIS_ASAMA_OLAP

```

    @HESAPTIPI          INT,
    @ASAMASIRA          INT,
    @ASAMAREF           INT,
    @SIPARISMIKTARI     FLOAT,
    @SIPARISHAZIROLMAZAMANI  FLOAT,
    @SIPARISBITISASAMA   FLOAT OUTPUT,
    @SECILENMAKINAREF    VARCHAR(12) OUTPUT
AS

```

```

DECLARE    @MAKINAREF VARCHAR(12),
           @SIPARISBITISMAKINA FLOAT

```

```

SET    @SIPARISBITISASAMA=990000000

```

```

DECLARE    CURSOR_MAKINA CURSOR FOR
SELECT MACHINEREF FROM TBL_MACHINEDEFINATIONS WHERE
STATIONREF=@ASAMAREF AND STATUS=1 AND MACHINEGROUPREF<>19 ORDER BY
MACHINEREF--19:GECICI MAKINA OLDUĞU İÇİN ÇIKARTILDI

```

```

OPEN CURSOR_MAKINA

```

```

FETCH NEXT FROM CURSOR_MAKINA
INTO @MAKINAREF

```

```

WHILE @@FETCH_STATUS=0
BEGIN

```

```

    EXEC T_SIPARIS_BITIS_MAKINA_OLAP @HESAPTIPI,
                                     @ASAMASIRA,
                                     @ASAMAREF,
                                     @MAKINAREF,
                                     @SIPARISMIKTARI,
                                     @SIPARISHAZIROLMAZAMANI,
                                     @SIPARISBITISMAKINA OUTPUT

```

```

    IF(@SIPARISBITISASAMA>@SIPARISBITISMAKINA)
    BEGIN

```

```

        SET @SIPARISBITISASAMA=@SIPARISBITISMAKINA
        SET @SECILENMAKINAREF=@MAKINAREF

```

```

    END

```

```

--    SELECT @MAKINAREF,@SIPARISBITISMAKINA AS
MAKINABITIS,@SIPARISBITISASAMA AS ASAMA,@SECILENMAKINAREF AS SECILEN

```

```

    FETCH NEXT FROM CURSOR_MAKINA
INTO @MAKINAREF

```

```

END

```

```

CLOSE CURSOR_MAKINA
DEALLOCATE CURSOR_MAKINA

```

```

UPDATE T_GANT_OLAP SET AKTIF=1 WHERE ASAMAREF=@ASAMAREF AND
MAKINAREF=@SECILENMAKINAREF
GO

```

CREATE PROCEDURE T_SIPARIS_BITIS_ASAMA_VTYS

@HESAPTIPI	INT,
@ASAMASIRA	INT,
@ASAMAREF	INT,
@SIPARISMIKTARI	FLOAT,
@SIPARISHAZIROLMAZAMANI	FLOAT,
@SIPARISBITISASAMA	FLOAT OUTPUT,
@SECILENMAKINAREF	VARCHAR(12) OUTPUT

AS

DECLARE @MAKINAREF VARCHAR(12),
 @SIPARISBITISMAKINA FLOAT

SET @SIPARISBITISASAMA=990000000

DECLARE CURSOR_MAKINA CURSOR FOR
 SELECT MACHINEREF FROM TBL_MACHINEDEFINATIONS WHERE
 STATIONREF=@ASAMAREF AND STATUS=1 AND MACHINEGROUPREF<>19 ORDER BY
 MACHINEREF--19:GECICI MAKINA OLDUĞU İÇİN ÇIKARTILDI

OPEN CURSOR_MAKINA

FETCH NEXT FROM CURSOR_MAKINA
 INTO @MAKINAREF

WHILE @@FETCH_STATUS=0
 BEGIN

EXEC T_SIPARIS_BITIS_MAKINA_VTYS @HESAPTIPI,
 @ASAMASIRA,
 @ASAMAREF,
 @MAKINAREF,
 @SIPARISMIKTARI,
 @SIPARISHAZIROLMAZAMANI,
 @SIPARISBITISMAKINA OUTPUT

IF(@SIPARISBITISASAMA>@SIPARISBITISMAKINA)

BEGIN

SET @SIPARISBITISASAMA=@SIPARISBITISMAKINA

SET @SECILENMAKINAREF=@MAKINAREF

END

FETCH NEXT FROM CURSOR_MAKINA
 INTO @MAKINAREF

END

CLOSE CURSOR_MAKINA

DEALLOCATE CURSOR_MAKINA

UPDATE T_GANT_VTYS SET AKTIF=1 WHERE ASAMAREF=@ASAMAREF AND
 MAKINAREF=@SECILENMAKINAREF

GO

CREATE PROCEDURE T_SIPARIS_BITIS_MAKINA_OLAP

```

    @HESAPTIPI                INT,
    @ASAMASIRA                INT,
    @ASAMAREF                INT,
    @MAKINAREF                VARCHAR(10),
    @SIPARISMIKTARI          FLOAT,
    @SIPARISHAZIROLMAZAMANI  FLOAT,
    @SIPARISBITIS            FLOAT OUTPUT
AS
DECLARE

    @MAKINAKAPASITE FLOAT,
    @KALANURETIM FLOAT,
    @KALANSURE FLOAT,
    @MAKINAHAZIROLMA FLOAT,
    @ISLEMEBASLAMA FLOAT,
    @SIPARISURETIMZAMANI FLOAT,
    @ISLEMBITIS FLOAT

SELECT
    @KALANURETIM=ISNULL(SUM(TMIKTAR),0)-ISNULL(SUM(TURETILEN),0)
FROM
    T_TOPLAM_MAKINA_TIP
WHERE
    STATIONREF=@ASAMAREF AND
    MACHINEREF=@MAKINAREF AND
    status IN (0,1) AND
    ITEMTYPE='OZEL'

SET
    @MAKINAKAPASITE=ISNULL(DBO.T_KAPASITE_OLAP(@HESAPTIPI,@ASAMAREF,@M
    AKINAREF),0)
    IF(@MAKINAKAPASITE=0) SET @MAKINAKAPASITE=1
    SET @KALANSURE=@KALANURETIM/@MAKINAKAPASITE
    SET @MAKINAHAZIROLMA=@KALANSURE
    SET @ISLEMEBASLAMA=@SIPARISHAZIROLMAZAMANI
    IF (@SIPARISHAZIROLMAZAMANI<@MAKINAHAZIROLMA) SET
    @ISLEMEBASLAMA=@MAKINAHAZIROLMA
    SET @SIPARISURETIMZAMANI=@SIPARISMIKTARI/@MAKINAKAPASITE
    SET @ISLEMBITIS=@ISLEMEBASLAMA+@SIPARISURETIMZAMANI

INSERT INTO T_GANT_OLAP
    (ASAMASIRA,AKTIF,ASAMAREF,MAKINAREF,KAPASITE,MEVCUTKALAN,MEVCUTBAS
    LAMA,MEVCUTSURE,MEVCUTBITIS,YENIHAZIROLMA,YENIBASLAMA,YENISURE,YENI
    BITIS)
VALUES(@ASAMASIRA,0,@ASAMAREF,@MAKINAREF,@MAKINAKAPASITE,@KALANU
    RETIM,0.0,@KALANSURE,@MAKINAHAZIROLMA,@SIPARISHAZIROLMAZAMANI,@ISL
    EMEBASLAMA,@SIPARISURETIMZAMANI,@ISLEMBITIS)
SET @SIPARISBITIS=@ISLEMBITIS
GO

```

```

CREATE PROCEDURE T_SIPARIS_BITIS_MAKINA_VTYS
    @HESAPTIPI                INT,
    @ASAMASIRA                INT,
    @ASAMAREF                 INT,
    @MAKINAREF                 VARCHA(10),
    @SIPARISMIKTARI           FLOAT,
    @SIPARISHAZIROLMAZAMANI   FLOAT,
    @SIPARISBITIS             AS FLOAT OUTPUT
AS
DECLARE
    @MAKINAKAPASITE FLOAT,
    @KALANURETIM FLOAT,
    @KALANSURE FLOAT,
    @MAKINAHAZIROLMA FLOAT,
    @ISLEMEBASLAMA FLOAT,
    @SIPARISURETIMZAMANI FLOAT,
    @ISLEMBITIS FLOAT
SELECT
    @KALANURETIM=ISNULL((SUM(T.IEMiktar)-SUM(T.Uretilen)),0)
FROM
    (
SELECT
        MT.STATIONREF,
        MT.MACHINEREF,
        IE.status,
        (CASE UE.CLIENTREF WHEN 0 THEN 'STD' ELSE 'OZEL' END) AS URUNTIPI,
        ISNULL(IE.WEIGHT,0) AS IEMiktar,
        ISNULL((SELECT SUM(U.WEIGHT) FROM dbo.TBL_PRODUCTION U WHERE
        U.WORKORDERREF=IE.WORKORDERREF),0) AS Uretilen
FROM
        dbo.TBL_WORKORDERS IE LEFT OUTER JOIN
        dbo.TBL_MACHINEDEFINATIONS MT ON MT.MACHINEREF=IE.MACHINEREF
LEFT OUTER JOIN
        dbo.TBL_PRODUCTIONORDERS UE ON
        UE.PRODUCTIONORDERREF=IE.PRODUCTIONORDERREF
WHERE
        MT.STATIONREF=@ASAMAREF AND
        MT.MACHINEREF=@MAKINAREF AND
        UE.CLIENTREF<>0 AND
        IE.status IN(0,1)
    ) T
SET
    @MAKINAKAPASITE=ISNULL(DBO.T_KAPASITE_VTYS(@HESAPTIPI,@ASAMAREF,@M
    AKINAREF),0)
    IF(@MAKINAKAPASITE=0) SET @MAKINAKAPASITE=1
    SET @KALANSURE=@KALANURETIM/@MAKINAKAPASITE
    SET @MAKINAHAZIROLMA=@KALANSURE
    SET @ISLEMEBASLAMA=@SIPARISHAZIROLMAZAMANI
    IF (@SIPARISHAZIROLMAZAMANI<@MAKINAHAZIROLMA) SET
    @ISLEMEBASLAMA=@MAKINAHAZIROLMA
    SET @SIPARISURETIMZAMANI=@SIPARISMIKTARI/@MAKINAKAPASITE
    SET @ISLEMBITIS=@ISLEMEBASLAMA+@SIPARISURETIMZAMANI
    INSERT INTO T_GANT_VTYS
    (ASAMASIRA,AKTIF,ASAMAREF,MAKINAREF,KAPASITE,MEVCUTKALAN,MEVCUTBAS
    LAMA,MEVCUTSURE,MEVCUTBITIS,YENIHAZIROLMA,YENIBASLAMA,YENISURE,YENI
    BITIS)

```

```
VALUES(@ASAMASIRA,0,@ASAMAREF,@MAKINAREF,@MAKINAKAPASITE,@KALANU  
RETIM,0.0,@KALANSURE,@MAKINHAZIROLMA,@SIPARISHAZIROLMAZAMANI,@ISL  
EMEBASLAMA,@SIPARISURETIMZAMANI,@ISLEMBITIS)  
SET @SIPARISBITIS=@ISLEMBITIS  
GO
```


ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	Çetinyokuş, Tahsin
Telefon	0 312 231 7400
e-mail	tahsinc@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek Lisans	GÜ Fen Bil. Enst.	2003
Lisans	OGÜ Müh.Mim. Fak. Endüstri Müh. Böl.	2000
Lise	Y.Mahalle Teknik Lisesi	1994

İş Deneyimi

Yıl	Yer	Görev
2002-	GÜ Müh.Mim.Fak. Endüstri Müh. Böl.	Arş.Gör.

Yabancı Dil

İngilizce