

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HİZMET SEKTÖRÜNDE VERİ ANALİZİ
(İŞ ZEKASI) VE MODELLEMESİ**

Matematik Mühendisi Fatma ALKANAT

**FBE Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. Mesut ÖZGÜRLER (YTÜ)

İSTANBUL, 2011

İÇİNDEKİLER

	Sayfa
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖNSÖZ	viii
ÖZET	ix
ABSTRACT	x
1. GİRİŞ	1
2. İŞ ZEKASI.....	3
2.1 İş Zekasının Tanımı	4
2.2 İş Zekası Çözümlerinin Tarihi	5
2.3 İş Zekası Çözümlerinin Temel Özellikleri	5
2.4 İş Zekası Çözümlerinin Getirdiği Yenilikler	6
2.5 İş Zekası Çözümlerinde Kullanılan Teknolojiler	8
2.5.1 Veri Ambarı	8
2.5.2 ETL	8
2.5.3 OLAP	9
2.5.4 Raporlar	9
2.5.5 Veri Madenciliği	10
2.5.6 Anahtar Performans Kriterleri	11
2.5.7 Kurumsal Karneler ve Gösterge Panelleri	12
2.6 İş Zekası Çözümlerinin Geleceği.....	12
3. VERİ AMBARI	14
3.1 Veri Ambarının Tarihsel Gelişim Süreci	14
3.2 Veri Ambarının Kullanım Amaçları	15
3.2.1 Tek ve Benzersiz Bir Ortam Oluşturmak	15
3.2.2 OLTP Sistemlerinin Yükünü Hafifletmek.....	15
3.3 Veri Ambarının Ayrıntıları	16
3.4 Veri Ambarında Kullanılan Veri Türleri	18
3.4.1 İçsel Veri.....	18
3.4.2 Dışsal Veri	19
3.4.3 Metadata	19
3.5 Data Mart	21
3.5.1 Veri Ambarı ile Data Martlar Arasındaki Farklar	22
3.5.2 Data Martlara İhtiyaç Duyulan Durumlar.....	22
4. VERİ AMBARI TASARIMI.....	23
4.1 Analiz Süreci	23
4.2 Veri Modelleme Teknikleri	25

4.2.1	Varlık İlişkili Modeller	26
4.2.2	Boyutsal Model.....	26
4.2.2.1	Olgu Tabloları.....	27
4.2.2.2	Boyut Tablolar	28
4.2.3	Veri Ambarı Şemaları.....	29
4.2.3.1	Yıldız Şeması.....	29
4.2.3.2	Kar Tanesi Şeması	30
4.3	Veri Yakalama, Dönüştürme ve Yükleme.....	30
4.3.1	Veri Yakalama	31
4.3.2	Veri Dönüştürme	32
4.3.3	Veri Yükleme.....	32
5.	ÇEVİRİMİÇİ ANALİTİK SÜREÇ	33
5.1	OLAP Kavramının Tarihi	33
5.2	OLAP Teknolojisinin Özellikleri	33
5.2.1	Veri Küplerine Çok Boyutlu Yaklaşım	33
5.2.2	Karmaşık Hesaplamalar.....	34
5.2.3	Zaman Boyutu	34
5.3	OLAP Kuralları	35
5.4	Temel OLAP Kavramları	37
5.4.1	Hiyerarşi	37
5.4.2	Ölçüm	37
5.4.3	Seviye	38
5.4.4	Üye.....	38
5.4.5	Kesme ve Dilimleme İşlemleri	38
5.4.6	Detay Açma ve Detay Kapama İşlemleri	39
5.4.7	Pivot İşlemi.....	40
5.5	OLAP Depolama Yöntemleri	40
5.5.1	ROLAP ile MOLAP Arasındaki Farklar	41
5.6	OLAP ve OLTP Karşılaştırılması.....	43
6.	HİZMET SEKTÖRÜNDE İŞ ZEKASI UYGULAMASI.....	46
6.1	Sağlık Sektöründe İş Zekası ile Çözümlenebilecek Güncel Sorunlar	46
6.2	Business Intelligence Development Studio	48
6.3	Veri Ambarı Tasarımı.....	51
6.3.1	Boyut Tabloları.....	51
6.3.2	Olgu Tabloları.....	56
6.4	Veri Aktarım İşlem	57
6.5	Küp Tasarımı	67
6.5.1	Projenin Oluşturulması	67
6.5.2	Data Source Tanımlanması.....	68
6.5.3	Data Source View Tanımlanması	68
6.5.4	Küp Boyutlarının Oluşturulması	71
6.5.5	Küpün Oluşturulması.....	74
6.6	MS Excel 2007 ile OLAP Küp Raporlaması.....	77
7.	SONUÇLAR.....	83

KAYNAKLAR.....	85
EKLER	88
Ek 1 Tabloların Oluřturulma Detayları.....	89
Ek 2 Kurumun Strateji Haritası.....	93
ÖZGEÇMİŐ.....	94

KISALTMA LİSTESİ

BI	Business Intelligence
CEO	Chief Executive Officer
CRM	Customer Relation Management
DNA	Deoxyribonucleic Acid
ER	Entity Relationship
ERP	Enterprise Resource Planning
ETL	Extract Transform Load
GUI	Graphical User Interface
IT	Information Technology
KPI	Key Performance Indicators
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing Systems
RDBMS	Relational Database Management Systems
SSIS	SQL Server Integration Services
SSAS	SQL Server Analysis Services
SQL	Structured Query Language
VTYS	Veri Tabanı Yönetim Sistemi

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1	Sektörlere göre iş zekası kullanımı [1] 4
Şekil 2.2	İş zekası ETL süreci [5] 9
Şekil 3.1	Metadata yönetiminin grafiksel gösterimi [6] 20
Şekil 3.2	Data mart [7]..... 21
Şekil 4.1	Veri ambarı analiz etki piramidi (Stackowiak vd., 2007)..... 24
Şekil 4.2	Fiziksel veri modelleri (Reinschmidt ve Francoise, 2000) 25
Şekil 4.3	Olgu tablosunun yapısı (Reinschmidt ve Francoise, 2000) 27
Şekil 4.4	Boyut tablosunun yapısı (Reinschmidt ve Francoise, 2000) 28
Şekil 4.5	Satış analizi için yıldız şeması örneği (Bain vd, 2001)..... 29
Şekil 4.6	Satış analizi için kar tanesi şeması örneği (Bain vd., 2001) 30
Şekil 5.1	Zaman hiyerarşisi [12] 37
Şekil 5.2	Bir ürünün belirli bir tarihteki satış ölçümü [13]..... 38
Şekil 5.3	Kesme ve dilimleme işlemleri [14]..... 39
Şekil 5.4	Detay açma ve detay kapatma işlemleri [14]..... 39
Şekil 5.5	Rolap çözümü [15] 40
Şekil 5.6	Molap çözümü [15]..... 41
Şekil 5.7	Holap çözümü [15] 43
Şekil 5.8	Operasyonel ve analitik veri tabanlarının farkları(Reinschmidt ve Francoise,2000) 45
Şekil 6.1	Business intelligence development studio kurulumu 48
Şekil 6.2	BI uygulama geliştirme ekranı..... 49
Şekil 6.3	BI Solution Explorer 50
Şekil 6.4	FacilityDim tablosu 52
Şekil 6.5	HealthPlanTypeDim tablosu..... 52
Şekil 6.6	HealthPlanPayerDim tablosu..... 53
Şekil 6.7	OrgTypeDim tablosu 53
Şekil 6.8	OrgSubTypeDim tablosu 53
Şekil 6.9	ResourceTypeDim tablosu..... 54
Şekil 6.10	ResourceDim tablosu..... 54
Şekil 6.11	ServiceDim tablosu..... 55
Şekil 6.12	VisitTypeDim tablosu..... 55
Şekil 6.13	DateDim tablosu 56
Şekil 6.14	VisitFact tablosu 57
Şekil 6.15	VolumeFact tablosu 57
Şekil 6.16	Integration Services Project oluşturma 58
Şekil 6.17	Integration Service Solution Explorer penceresi 59
Şekil 6.18	Excel bağlantı yönetimi 60
Şekil 6.19	Facility.xls 60
Şekil 6.20	Veri dönüştürme işlemi..... 61
Şekil 6.21	Slowly changing dimension wizard..... 62
Şekil 6.22	Slowly changing dimension boyut seçimi 62
Şekil 6.23	ETL paketinin akış şeması..... 63
Şekil 6.24	Sequence container 64
Şekil 6.25	Execute SQL Task editörü..... 65
Şekil 6.26	Yeni alan ekleme ve dönüştürme ekranı..... 65
Şekil 6.27	Alan eşleştirmelerinin yapılışı 66

Şekil 6.28	Olgu tablosu için paket akışı.....	67
Şekil 6.29	OLAP projesinin nesneleri	68
Şekil 6.30	Küp için gerekli tabloların seçilişi.....	69
Şekil 6.31	Ziyaret için yıldız şema modeli	70
Şekil 6.32	İşlemler için kar tanesi şema modeli	71
Şekil 6.33	Boyut yaratma sihirbazı.....	72
Şekil 6.34	Date boyutu için tablo ve key alan seçimi	72
Şekil 6.35	Tarih boyutunun detayları.....	73
Şekil 6.36	Hiyerarşi ilişkileri ve boyut verilerinin görünümü	74
Şekil 6.37	Küp için gerekli olgu ve boyut değerlerinin belirlenişi	74
Şekil 6.38	Boyut ile ölçümler arasındaki fiziksel ilişkiler	75
Şekil 6.39	Küp verilerinin incelenme ekranı	76
Şekil 6.40	Excel 2007 ile SQL Server 2008 R2 Analysis Services'e bağlantı	77
Şekil 6.41	Özet tablo ve özet grafik sihirbazı	78
Şekil 6.42	Verilerin özet tablo biçimindeki gösterimi	78
Şekil 6.43	Ziyaret tiplerine göre işlem anlizleri raporu	79
Şekil 6.44	Hastaların sağlık güvencesi dağılımı analizi raporu	80
Şekil 6.45	İşlem ve ziyaret bilgilerinin trend analiz raporu	81
Şekil 6.46	Personel bazlı performans analizi raporu	82

ÖNSÖZ

Tez çalışmam boyunca benden desteğini, bilgisini ve sabrını esirgemeyen saygıdeğer hocam Prof. Dr. Mesut Özgürler'e, görüş ve yönlendirmeleri için sayın hocam Semih Önüt'e ve Arş. Gör. Şenim Özgürler'e, değerli yardımları için eşime ve bugünlere gelmemi sağlayıp hep yanımda olan sevgili aileme teşekkürü bir borç bilirim.

Bana çalışırken yüksek lisans yapabilme imkanı tanıyan görev aldığım şirket ve ekiplere, ayrıca eğitimime burs katkısı sağlayan bilim insanı destekleme programına teşekkür ederim.

Ocak, 2011

ÖZET

Sağlık hizmetlerinde beklentilerin karşılanması için; gelişen teknolojinin, bilimsel bilgi birikiminin, insan sağlığını korumak, tedavi ve rehabilite etmek amacıyla, hızlı, verimli ve doğru olarak uygulanması gerekmektedir.

Sürekli olarak artış gösteren operasyonel veriler üzerinde istatistiksel anlamda sorgular çalıştırmak neredeyse imkansız hale gelmiştir. Sadece küçük zaman aralıkları içinde parçalanmış sorgularla sonuca ulaşılabilir. Bu bağlamda, İş zekası uygulamaları, kullanıldıkları her sektörde olduğu gibi, sağlık sektöründe de önemini gün geçtikte artırmaktadır. Söz konusu uygulamalar, özetlenmiş ve gruplanmış veriler üzerinde, hızlı ve kolay erişimi sağlayan veri ambarı ve OLAP teknolojilerini kullanarak raporlama ve istatistiki veri ihtiyacını etkili bir şekilde karşılamaktadır.

Operasyonel verilerin, bu teknolojiler yardımıyla ETL sürecinden geçirilerek, veri ambarında saklanacak yapıya dönüştürülmesi ve mevcut hatalarından arındırılması sağlanarak, kaliteli veriler, veri ambarı ortamına aktarılır. OLAP teknolojisi ile çok boyutlu küpler yaratılarak, verilerin daha kolay ve hızlı sorgulanmasını sağlayan modellemeler yapılabilir. Böylelikle temiz ve kaliteli veriler üzerinde yapılacak veri madenciliği uygulamaları ile veriden, yeni ve doğru bilgilere ulaşılır.

Gerçekleştirilen uygulama, özel bir sağlık kurumunda farklı kaynaklarda bulunan verilerin merkezi bir veri ambarında toplanması, toplanan bu verilerin çok boyutlu analize uygun bir yapıda modellenmesi, daha sonra işlenerek özellikle yöneticiler ve analistler tarafından ihtiyaç duyulan ve karar alma sürecini etkileyen raporlamaların MS Excel 2007 üzerinden sunulması adımlarından oluşan, uçtan uca bir iş zekası uygulamasıdır.

Anahtar Kelimeler: İş zekası, veri ambarı, OLAP, ETL, sağlık sektörü

ABSTRACT

In order to provide the expectations for healthcare services, while protecting the human health and treating the diseases, it is required to implement the developing technology, scientific knowledge, with the fast and efficient methods applying in the right way.

By the day, business intelligence takes an important role in medical information technology. To run statistical queries on continuously increasing operational data has almost been impossible. The segmented queries dedicated for small time periods might be the only way to achieve the result in such system. Business intelligence applications use the data warehouse and OLAP technology which let the users access the summarized and grouped data in a fast and an easy way.

With the help of the technologies mentioned above, after the operational data is passed through the ETL process and converted into the suitable format just for storing, while providing the error free and the quality data in data warehouse environment. It is provided some models let the user query the data in easy and fast manner via creating multidimensional cubes with the help of OLAP technology. Thus, it might be possible to access the new information using clean and the high quality data in data mining applications.

The application developed is an end-to-end business application that consists of the following steps; gathering the data of a private health institution located in different storages into a central data warehouse, modeling this data suitable for multi-dimensional analysis, then processing it and submitting the decision support reports MS Excel 2007 form which are required especially by managers and analysts.

Keywords: Business intelligence, data warehouse, OLAP, ETL, healthcare.

1. GİRİŞ

Günümüzde şirketler veri miktarlarının artışıyla birlikte bu verilerden özet almak, veriyi görüntülemek ve karar vermek için raporlama ihtiyacı duymaktadırlar. Bu ihtiyaç bilgi sistemlerine sahip olmayan şirketlerde en basit haliyle şirket çalışanlarının yazılı olarak veri temini ve toplanan verilerin hiyerarşik olarak üst yönetime raporlanması ile karşılanır. Bilgi sistemleri kullanan küçük ölçekli şirketler ise raporlama ihtiyaçlarını operasyonel sistemlerden aldıkları raporlarla karşılayabilmektedirler.

Raporlar üzerinde analiz yaparak hızlı karar verme ihtiyaçlarının artması ve raporlama için ayrılan kaynakların ihtiyaçlara cevap vermemeye başlamasıyla, şirketler yeni çözüm arayışlarına girmektedirler. İş zekası çözümleri de, şirketlerin daha hızlı ve doğru karar alabilmeleri için; verilerin toplanmasını, depolanmasını, analiz edilmesini, verilere belirli yetkiler dahilinde erişilmesini, planlama yapılmasını, stratejilerin belirlenmesini, kritik yönetim kararlarının verilmesini sağlayan uygulama ve teknolojilerdir.

İş zekası çözümlerinin kullanıldığı sektörden biri de, insan yaşamındaki kritik yeri nedeni ile teknolojik gelişmelerin yakından takip edildiği ve uygulandığı sağlık sektörüdür. İş zekası çözümlerinin dünyada birinci sırada gösterilen teknolojiler olduğu göz ardı edilmemeli ve ülkemizdeki sağlık kurumları tarafından yeterince önemsenmelidirler. Böylelikle sağlık kurumlarının karmaşık yapısı ile çevreleri arasındaki denge sağlanır, insanlara sağladıkları hizmet kalitesi yükselir, karlılık ve verimlilikleri artar.

Bu bağlamda iş zekası teknolojisini, sağlık sektörü üzerine modelleme yaparak ele alan çalışmanın ikinci bölümünde iş zekası teknolojisi hakkında kuramsal bilgiler verilmiş, üçüncü ve dördüncü bölümlerinde iş zekası sistemlerinin temelini oluşturan veri ambarı ve onun temel kavramları detaylı olarak incelenmiş, beşinci bölümde ise çok boyutlu analiz yapabilmeyi sağlayan OLAP mimarisi anlatılmıştır.

Altıncı bölümde, sağlık sektörünün güncel sorunlarına iş zekasının getirdiği çözümlerin üzerinde durularak uygulamaya başlanmıştır. Uygulamada hastane yetkilileri ile yapılan analiz çalışmaları sonucu projenin kapsam ve amaçları belirlenmiş ve kurumun

ilk fazda ihtiya duyacađı raporlar karřılıklı fikir alışverişleri ile tespit edilmiştir. Bir sonraki aşamada doğru ve net bilgiye ulaşabilmek amacıyla, kullanıcı hatası ve program eksikliklerinden kaynaklanan hatalı ve eksik veri girişleri SQL ifadeleri yardımıyla temizlenmiş ve veriler SQL Server Integration Services yardımıyla veri ambarına taşınmıştır. Elde edilen veriler üzerinde ok boyutlu analiz işlemlerinin yapılabilmesi için OLAP küp yapısı SQL Server Analysis Services yardımıyla oluşturulmuştur. Kullanıcıların, OLAP küplerine daha kolay, esnek ve grafik destekli olarak ulaşabilmeleri için raporlama aracı olarak Excel kullanılmıştır.

Yedinci ve son bölümde ise alışma ile elde edilen sonuçlara yer verilmiş olup, alışmanın devamı niteliğini taşıyabilecek önerilerde bulunulmuştur.

2. İŞ ZEKASI

İşletmelerin yaşam döngüleri her gün yüksek miktarda veri oluşturmaktadır. Bu verilerin saklanması, analiz edilmesi ve işletmelerin bu verilerden topladıkları bilgiler ile kendilerini yönlendirmeleri gerekir. İşletmelere bu yolda rehberlik edecek en güncel teknoloji ise iş zekasıdır.

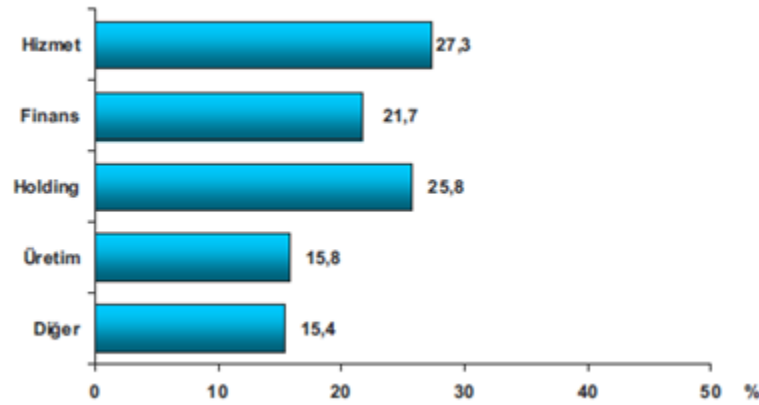
Geçmiş dönemlerde arz kısıtlı ve rekabet sınırlı olduğu için, üretilen her ürün değerli iken, son yıllarda arzın talebe göre daha hızlı artması ve rekabet nedeniyle müşteri daha değerli olmuştur. Bu değişim sonucunda ürün ve üretim planlamaları ile müşteri istek ve ihtiyaçlarının analiz edilmesi ön plana çıkmıştır. Hızla gelişen teknoloji, müşteri taleplerinin de aynı hızda farklılaşabilmesine yol açmıştır. Bütün bunların sonucunda, müşteri eğilimlerinin anlık fotoğrafını çekmek ve bu doğrultuda uzun dönemi, örneğin gelecek iki yılı planlamak oldukça riskli olmuştur. İş zekası burada devreye girmekte, müşterilerin farklılaşan eğilimleri doğrultusunda, üretim planlarında sürekli değişimi sağlamaktadır. Bu atıl stok maliyetlerinin düşmesi, verimlilik ve en önemlisi karlılık demektir.

Müşteri taleplerine göre üretim yapmak bir şirketin sürekliliği, gelir ve kar artışı için tek başına yeterli değildir. Sadece talebi doğru anlayan şirketler pazar ortamında farklılaşabilmektedirler. Talebi karşılayan ve müşteriyi memnun eden firmalar pazardaki fiyatlandırmada daha fazla söz sahibi olmaktadır. Yönetim uzmanı Kash'in (2002), "The Law of Demand and Supply" adlı kitabında talep stratejisini uygulayan şirketlerin karlarını arttıracak dört avantaja sahip oldukları belirtilmektedir. Bu avantajlar şunlardır:

1. Müşteriye tam olarak istediğini satmak (Sürekli müşteri ilişkileri yönetim programları ile veri toplamak, analiz etmek).
2. Ürünleri, rakip firma ürünlerinden farklılaştırma olanağına kavuşmak (Piyasa ürün araştırması).
3. Talebi kısıtlı olan ürünleri, pazardan çekerek maliyetleri düşürme imkanı yakalamak (Satış grafikleri analizi).
4. Ürün ya da servisleri çok uygun fiyatlarla müşteriyle buluşturma olanağı yakalamak (Piyasa fiyat analizi).

Her CEO bu dört avantaja sahip olmak isteyecektir. Bunu sağlayabilecek olanlar ise iş zekası ile elindeki doğru verileri uygun yazılımlarla karar önerilerine dönüştürebilenler olacaktır.

İş zekası, dünyada son 15-20 yıldır en fazla ilgi duyulan kavramların başında gelmektedir. Ülkemizde iş zekasının gerekliliği geçtiğimiz 1,5–2 yılda daha iyi anlaşılmış olup, iş zekasına olan ilgi gün geçtikçe artış göstermektedir. İşletmelere karlılık, müşteri ve çalışan memnuniyeti, performans, kalite, verimlilik gibi yönlerden avantajlar sağlayan iş zekası, bütün sektörlerde uygulanabilmektedir. Ülkemizde sektörlerde göre iş zekası çözümlerinin kullanım oranları Şekil 2.1’de gösterilmiştir.



Şekil 2.1 Sektörlere göre iş zekası kullanımı [1]

2.1 İş Zekasının Tanımı

İş zekası için yapılmış birçok tanım olmakla birlikte, en basit olarak; “Kurumların daha hızlı ve doğru karar alabilmeleri için; verilerin toplanmasını, depolanmasını, analiz edilmesini, verilere belirli yetkiler dahilinde erişilmesini, planlama yapılmasını, stratejilerin belirlenmesini, kritik yönetim kararlarının verilmesini sağlayan uygulama ve teknolojilerin bütünüdür.” şeklinde ifade edilebilir.

Bir diğer tanıma göre; “Çalışanların işle ilgili verilere kolayca erişebilmeleri için kullanılan veri tabanları, karar destek sistemleri ve bütünleşik işlemlerden oluşan bir mimaridir.” (Moss ve Atre, 2003). Bu tanıma göre, iş zekası sadece bir ürün ya da sistemden oluşmaz, bütünleşik bir yapıdır.

2.2 İş Zekası Çözümlerinin Tarihi

Bilginin saklanması ve değerlendirilmesi sadece günümüz insanının ortaya koyduğu bir fikir değildir. Bilgisayarların insan yaşamında daha büyük yer edinmesi verilerin depolanmasında büyük bir devrim yarattı. Elle yapılan pek çok işlem otomasyon sistemlerine geçirildi. Otomasyon sistemleri hesaplamalardaki doğruluk oranlarını arttırıp, işlemleri daha hızlı ve güvenilir hale getirdi.

1983 yılında ilk hesap çizelgesi olarak Lotus piyasaya sürüldü. Bu gelişim ile ortaya çıkan hesaplama tabloları, kişisel bilgisayarları, kullanıcıların kendi analiz ve hesaplarını yapabilecekleri şekilde güçlendirdi. Kalan tek problem, veriye erişim hızının düşük olmasıydı.

1984'lerin sonunda istemci-sunucu sistem fikri doğdu. Bu fikrin temelinde, verinin asıl yerinde tutulması gerektiği düşüncesi vardı. Veri, farklı bilgisayarlarda değişik formatlarda saklanıyordu ve bulunduğu kaynaktan alınıp başka bilgisayarlara aktarılması ciddi maliyet gerektiriyordu. Zamanla veri tabanlarında standart yaratılması zorunlu hale geldi. Bu standartlaştırma çalışmaları, iş zekası teknolojisi için zemin hazırlamış oldu (Biere, 2003).

Daha sonra üst yönetim için verileri bütünleşik ve özet halde sunan karar destek sistemleri geliştirildi. Zaman içinde bu sistemlerde fark edilen görsel yetersizlikler, kullanım zorlukları ve uygulamalar arasındaki uyumsuzluklar yeni bir sistem ihtiyacını yarattı. Bu ihtiyaçlar sonucu iş zekası çözümleri uygulamaya geçirildi.

2.3 İş Zekası Çözümlerinin Temel Özellikleri

Dünya genelinde pek çok sektöre BI çözümleri sunan Business Objects, iş zekası çözümleri ile ilgili temel özellikleri 4 maddede sıralamaktadır [2].

- Bilgiye tek ortak noktadan erişim: İş zekası kullanan işletmelerde, veriler ortak bir veri tabanında saklanır. Herhangi bir anda bilgiye ihtiyaç duyan kişiler, ister şirket içinden ister uzaktan, çeşitli arayüzler veya internet aracılığıyla sisteme bağlanıp, istediği bilgiye kolayca erişebilir. Kullanıcının bu işlem sırasında teknik detayları bilmesine gerek yoktur.
- İşletmenin bütün bölümlerinde kullanılabilir olması: İş zekası çözümleri

yaygınlaşmadan önce, işletmedeki her bir bölüm kendi içinde verileri saklamaktaydı. Bölümlerin bunun için kullandıkları yazılımlar ve araçlar birbirinden farklı olabiliyordu. Bu durumda bölümler arasında ortak bir dil konuşmak mümkün olmuyordu. İş zekası, işletmedeki bütün bölümlerin verilerini bir araya toplayarak, istenilen bilgiye daha kolay ve çabuk ulaşılabilmesini sağlamıştır.

- Ortaya çıkan sorulara anında cevap verebilmesi: İşletmelerde gün içinde çok sayıda rapor hazırlanmaktadır. Raporların incelenmesi sırasında ortaya yeni sorular çıkabilmektedir. Her sorunun cevabı için yeni araştırmalar yapmak, zaman ve uğraş gerektirmektedir. Raporu okuyan kişilerin akıllarına takılan sorulara dinamik olarak cevap alabildikleri uygulamalar esnek çözümler olacaktır. Business Objects'e göre, insanlar zamanlarının %80'ini standart raporlar için, geri kalan %20'sini ise bu sonradan ortaya çıkan ve raporun kendisinde cevabı bulunmayan sorularla ilgili ek bilgi alabilmek için harcamaktadırlar. İş zekası bu %20'lik zamanı en verimli şekilde kullanmayı amaçlar. Kullanıcılar iş zekası sistemlerinde, basit sorgulamalar yaparak aradıkları cevaplara ulaşabilmektedir.

- İnternetin olanaklarından yararlanabilmesi: İş zekası sistemlerine belirli güvenlik limitleri içinde uzaktan erişim mümkündür. Şirket dışındaki bir kullanıcı, şirketin bilgi ağına bağlanıp istediği verilere ulaşabilmektedir. Örneğin; bir tedarik firması, mal sattığı işletmenin satış rakamlarını bu sayede inceleyebilmektedir.

2.4 İş Zekası Çözümlerinin Getirdiği Yenilikler

İş zekası çözümleri işletmelere sağladığı yenilikler sayesinde hızlıca yaygınlaşmıştır. İş zekasının kapsamı organizasyonlarda ihtiyaçlara göre geniş ya da sınırlı olabilir. Kullanıcı sayısı ve kullanıldığı bölümler farklılık gösterebilir (Çağıltay, 2010).

İşletmeler için sahip olunan veriler ve bunların işlenmesiyle ortaya çıkan bilgiler, işin durumunu, performansını ve işletmenin geleceğini anlamak, müşterileri ve çözüm ortaklarıyla ilişkilerini güçlendirmek, operasyonlarını iyileştirmek açısından büyük önem taşır. İş zekası, işletmenin içinde ve dışında ortaya çıkan bütün verileri bir depoda toplar. Bütün bölümler aynı depodan veriyi kullanır ve veriyi güncelledikten sonra yine aynı yere yazarlar. Bölümler arasındaki bilgi tutarsızlıkları da bu sayede önlenir.

İş zekasının bir diğer özelliği ise bilginin yaratılabilmesi için veriyi analiz edip yorumlayabilme imkanı tanınmasıdır. Bunun için sorgulama, raporlama ve çok boyutlu analiz gibi karar destek araçlarından yararlanır. Ortaya çıkarılan bilgiler, detaylı ya da özet raporlarla kullanıcılara sunulur. Kullanıcılara tek bir veri tabanını ve mimariyi kullanarak, istedikleri biçimde raporlar hazırlama imkanı tanır. İş zekasının sunduğu rapor formatları, hücrelerden grafiklere, tablolardan haritalara kadar geniş bir yelpazede değişir [3].

Ortaya çıkarılan bilgiler, internet üzerinden bütün yöneticilere, iş ortaklarına, bilgi kullanıcılarına güvenli bir şekilde aktarılabilir. Böylelikle işletme içsel ve dışsal iletişimini güçlendirir, sürekli gelişmeye açık olur. Yönetici, istediği kritere göre işletmenin performansını görüntüleyebilir (Howson, 2008).

Yöneticiler, işletmedeki ilerleyişin kendi planları çerçevesinde olup olmadığını sorgulayıp, rakip firmalarla kıyaslamalar yapabilir. İş zekası uygulamaları işletmeye kazanç sağlayacak alanların bulunmasını kolaylaştırır. Bu sayede işletme önceliklerini belirler. Kaynaklarını ve çalışmalarını bu alanlara yöneltir.

Diğer önemli özellikleri, esas sebebi bulma analizleri ve uyarı mekanizmalarıdır. Kullanıcılar günlük raporlara bakarken farkettileri problemlerin nedenlerini öğrenmek isteyebilirler. Bu gibi durumlarda detaylı incelemeler yapmak için iş zekası geniş olanaklar sunar. Kullanımı kolay arayüzlerle veriler arasında gezinmek, veri tabanlarına inmek ve veriye farklı açılardan bakmayı sağlayacak sorgulamalar yapmak mümkündür. Bütün bunlar, iş zekası uygulamalarında raporlama ve analiz sistemlerinin ileri biçimde bütünleşmiş olmasıyla sağlanır.

Sistemde herhangi bir anda meydana gelen değişimler için kullanıcıları ve üst yöneticileri uyaran mekanizmalar oluşturulabilir. Dinamiklerde meydana gelen bir değişim, anında ilgili kişilere yönlendirilebilir. Örneğin; üretim hatlarının standart çalışma işleyişinde meydana gelen bir arıza, uyarı amaçlı elektronik mektuplarla ya da smsler ile ilgili kişilere otomatik olarak iletilebilir [4]

İş zekası sistemleri, eldeki büyük miktarda veriyi analiz ederken tahminleme algoritmalarından da faydalanır. Böylece işletmenin anlık görüntüsünü vermek dışında gelecekle ilgili planlar yapmasına da yardımcı olur.

Bütün bu yeni özellikler, büyük firmalardan küçük ve orta büyüklükteki işletmelere kadar bir çok işletmenin sistemlerini iş zekasına taşıma kararı almalarını sağlar.

2.5 İş Zekası Çözümlerinde Kullanılan Teknolojiler

İş zekası sistemlerine bakıldığında, şu ana teknolojiler göze çarpar: Veri ambarları, analiz araçları ve raporlama araçları.

Veri ambarları, veri tabanları ve yapılandırılmamış kaynaklardan gelen verileri toplayan ve analizler için bütünleyen yapılardır. Analiz araçlarına örnek, veri madenciliği araçları olabilir. Bunlar veriyi analiz eden ve sonuçlar çıkaran araçlardır. Raporlama araçları, analistler dışında bilgiyi kullanan kimseler için ihtiyaca göre detaylı ya da özet bilgileri göstermek üzere tasarlanmış araçlardır (Azvine vd., 2005).

2.5.1 Veri Ambarı

Veri ambarı, farklı operasyonel sistemlerden iş verilerini çıkaran, onları tutarlı bir hale dönüştüren ve analiz için dağıtan iş zekasının temel elemanıdır (Howson, 2008).

Veri ambarı, analiz amaçlı veriyi toplayan ve organize eden, bu sayede yönetimin işle ilgili bilgilere erişmesini ve analiz edilmesine olanak sunan bir teknolojidir (Reinschmidt ve Francoise, 2000). İş zekası sistemlerinin temelini oluşturan veri ambarı teknolojilerine ilerleyen bölümlerde ayrıntılı olarak değinilecektir.

2.5.2 ETL

ETL sistemleri çeşitli veri kaynaklarından (OLTP, ERP, CRM, Excel dosyaları, csv dosyaları vb.) verileri çeker, bunları temizler ve birbirleriyle uyumlu hale getirir, sonra da bu verileri merkezi veri ambarına yapılacak sorgulamalara cevap verecek şekilde yükler. ETL sürecinin veri ambarı yapısı içindeki yeri Şekil 2.2’de görüntülenmektedir.



Şekil 2.2 İş zekası ETL süreci [5]

2.5.3 OLAP

Çevrimiçi analitik süreç için şöyle bir tanımlama yapılabilir: Varolan veriler üzerinde bir takım hesaplamalar ve dönüştürmeler yaparak, işle ilgili bilgilerin ortaya çıkmasını sağlayan analitik işlemci teknolojisidir (Moss ve Atre, 2003).

OLAP, veri ambarlarının tamamlayıcısı olarak düşünülür ve bütünleşik verilerin çok boyutlu analiz edilmesinde kullanılır. Bu sayede veriye farklı açılardan bakarak hızlı, tutarlı ve etkileşimli bilgiler elde edilir. Veri ambarı, veriyi saklayan ve yöneten teknolojidir. OLAP ise, bu veriyi stratejik bilgiye dönüştürür. Ortaya atılan bir hipotezin doğruluğunu çok yönlü olarak araştırır. OLAP teknolojisinde veriye ulaşmak için kullanılan basit işlemler dışında, karmaşık hesaplamalar, zaman serileri ve modellemeler de yer alır (Reinschmidt ve Francoise, 2000). OLAP, bütün bu işlemleri gerçek zamanlı olarak yapar. Desteklediği karmaşık hesaplamalarla karar verme sürecinde yöneticilere kolaylık sağlayacak bilgiler yaratır. Örneğin; hareketli ortalama gibi trend algoritmalarını kullanarak satış tahminlerine yardımcı olur (Howson, 2008).

2.5.4 Raporlar

Raporlar, iş zekası sisteminin en basit elemanlarıdır. İşle ilgili bilgilerin karar vericilere ulaştırılmasında kullanılırlar. Raporlar analiz sırasında elde edilen bilgilerin işletme

içinde paylaşılması için önemlidir. Yerel alan ağlarının yaygınlaşmasıyla birlikte raporların ulaşılabilirliği artmıştır. Raporlar elektronik postalar, internet, sesli mesajlar ve el bilgisayarları ile iletmeye başlanmıştır. Sadece yazıdan oluşan uzun raporların anlaşılması güç olduğu için, iş zekası sistemlerinde raporlar; grafikler, tablolar ve özetlerle zenginleştirilir.

Veri ambarında ya da operasyonel sistemlerde bulunan verinin raporlara yansıtılması için arada sorgulama araçları çalışır. Bu araçlar veri kaynaklarından aldıkları verileri, raporlama araçlarına ileterek onlar tarafından sunulabilir hale getirilmesine yardımcı olur. Bu sayede kullanıcılar fiziksel veri tabanındaki karmaşık yapıyı bilmek zorunda kalmazlar, sorgulama araçlarına anlaşılabilir sorgu cümleleri girerek kendilerine dönen sonuçlarla ilgilenirler. Sorgulama araçları, veri tabanı ile raporlama araçları arasında bir arayüz olarak da düşünülebilir. En bilindik ve en yaygın sorgulama dili SQL'dir. Hangi veri tabanı çözümünün kullanıldığından bağımsız olarak bütün ortamlarda çalışmaktadır (Howson, 2008).

2.5.5 Veri Madenciliği

Veri madenciliği, eldeki verilerden üstü kapalı, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilgilerin çıkarılmasıdır. Veri madenciliği kümeleme, veri özetleme, değişikliklerin analizi, sapmaların tespiti gibi teknik yaklaşımları içerir. Başka bir deyişle, veri madenciliği, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir.

Temel olarak veri madenciliği, geniş veri setleri arasındaki desenlerin ya da düzenin, verinin analizi ve yazılım tekniklerinin kullanılması ile ilgilidir. Veriler arasındaki ilişkiyi, kuralları ve özellikleri belirlemekten bilgisayar sorumludur. Amaç, daha önceden fark edilmemiş veri desenlerini tespit edebilmektir.

Veri madenciliğini istatistiksel yöntemler serisi olarak görmek uygun olabilir. Ancak veri madenciliği, geleneksel istatistikten birkaç yönüyle farklılık gösterir. Veri madenciliğinde amaç, kolaylıkla mantıksal kurallara ya da görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır.

Veri madenciliği ve bilgi keşfi, özellikle elektronik ticaret, bilim, tıp, iş ve eğitim alanlarındaki uygulamalarda yeni ve temel bir araştırma sahası olarak ortaya çıkmaya başlamıştır. Web üzerinde filtrelemeler, DNA sıraları içerisinde genlerin tespiti, ekonomideki eğilim ve düzensizliklerin tespiti, elektronik alışveriş yapan müşterilerin alışkanlıkları gibi karar verme mekanizmaları için önemli bulgular elde edilebilir.

Yukarıda da özetlendiği gibi veri madenciliği çalışmaları günümüz bilgi toplumunda kritik bir alan olmaya başlamıştır. Avrupa ve ABD de veri madenciliği konusunda birçok araştırma grubu kurulmuş ve kurulmaktadır.

2.5.6 Anahtar Performans Kriterleri

Anahtar performans kriterleri, bir işletmenin içinde bulunduğu durumu değerlendirmek amacıyla incelenen ve kurumsal performans yönetiminde kullanılan ölçütlerdir. İşlerin iyi mi kötü mü gittiği ve eğilimlerin ne yönde olduğu hakkında bilgi verirler. İşle ilgili verilerin istenilen değerde olup olmadığı KPI'larla ölçülür. Dolayısıyla her KPI, gerçekleşen durumu gösteren sayısal bir değerle birlikte, planlanan değeri de tutar. KPI'ların raporlarda gösterimi sırasında, gerçek ve planlanan değerleri ile birlikte, planlanan değere olan uzaklık ve yakınlıklarını ifade eden semboller de kullanılır. Bunlar bazen trafik ışıkları gibi kırmızı, sarı, yeşil renklerle, bazen termometre gösterimi ile bazen de yüz ifadeleri ile anlaşılır hale getirilir.

İşletmeler için en fazla dikkat gerektiren konu KPI'ları belirlemektir. Bunun için kullanılabilecek ilk strateji, çalışanların en çok hangi verilere ihtiyaç duyduklarının belirlenmesidir. KPI'lar için belirleyici olabilecek diğer bir kaynak, üst yönetimin hangi veriler için raporlar istediği, hangi ölçütleri sıkça takip ettiğidir. Her işletmenin kendisi için önemli saydığı kriterler farklı olacağından, her işletme, her bölüm ve hatta her kullanıcı için seçilecek KPI'ların da birbirinden farklı olacağı açıktır (Utley, 2008).

İş zekası sistemleri tamamlanmış süreçlerden oluşmazlar. İhtiyaçlara ve işletmeye katılan yeni sistemlere bağlı olarak sürekli gelişir ve büyürler. Bu da zamanla yeni KPI'ların ortaya çıkmasına neden olabilir (Andersson vd., 2008).

2.5.7 Kurumsal Karneler ve Gösterge Panelleri

Karneler ve gösterge panelleri, performans verilerini görüntülemek için kullanılan iş zekasının iki önemli elemanıdır. Karneler, önceden belirlenen kriterler için elde edilen gerçek değerlerle planlanan değerleri karşılaştırmak için kullanılırlar. Gösterge panelleri ise, bilgiyi sunarken karneler ve diğer görsel araçları bir arada kullanırlar. (Andersson vd., 2008).

Karneler için gidişatı ile ilgili bilgi vermek amacıyla hazırlanan ve KPI'lardan oluşan yapılardır. İşletmenin bütünü için genellikle tek karne bulunsa da, her bir bölüm ya da yönetici için farklı karneler de hazırlanabilir (Kaplan ve Norton, 2006). Karneler özellikle karar vericiler için, fazla detaya girmeden işe bakabilmek adına çok önemlidir. İşletme için en önemli karneler kurumsal karnelerdir. Kurumsal karneler, ilk olarak 1992'de Robert Kaplan ve David Norton tarafından adından söz edilen, işletmenin başarısını ölçmek için farklı bakış açıları ve ölçütler belirleyen karnelerdir.

Gösterge panelleri, bir ya da daha çok hedefe ulaşmak için ihtiyaç duyulan bilgileri tek bir ekran üzerinde bir bakışta görebilmek amacıyla hazırlanmış, birleştirilmiş ve düzenlenmiş araçlardır. Bunlar, farklı kaynaklardan topladıkları bilgileri, anlaşılması kolay ve hızlı bir şekilde kullanıcıya sunarlar. Sunum yöntemi bir rapor ya da excel dosyasındaki bir grafik olabileceği gibi, bir pdf dökümanı ya da elektronik mektup da olabilir (Stackowiak vd., 2007).

2.6 İş Zekası Çözümlerinin Geleceği

İş zekası uygulamalarının yakın zamanlara kadar pahalı olmasının önemli sebeplerinden biri işlem gücü ve donanım olarak yüksek kaynaklar gerektirmesiydi. Donanım yeteneklerinin gelişmesi ve ucuzlamasıyla birlikte kullanılan araçların ucuzlaşıp yaygınlaşması ve iş zekası alanındaki bilgi birikiminin artmasıyla, iş zekası uygulamaları artmaya başlamıştır. Gartner'ın geçmiş yıllarda yaptığı araştırmaların sonucunda, IT yöneticilerinin daha önce en fazla yatırımı güvenlik alanına yaptığını, son yıllarda ise en çok iş zekası uygulamalarına yatırım yapacakları vurgulanmıştır.

Gartner tarafından 20 - 22 Ocak 2009'da Hollanda'da yapılan iş zekası yıllık buluşmasında şu tahminlere yer verilmiştir [1]:

- 2012'ye kadar, dünyadaki en büyük 5000 çok uluslu firmanın üst düzey yetkililerinden %35'i, kendi sektörlerindeki önemli değişiklikler ve piyasalarla ilgili yanlış kararlar alacaklar,
- İş zekası konusunda harcanacak bütçelerin en az %40'ı 2012 yılına kadar, bilgi teknolojileri dışındaki iş birimleri tarafından kontrol edilecek,
- Kurumların ancak %20'si, 2010 yılına kadar sektörlerine özgü bir analitik uygulamaya, iş zekası çözümü içinde standart bir bileşen olarak ulaşmış olacak,
- 2012 yılına kadar analitik yazılımların %33'ü, iş süreçlerine uyarlanıp iş zekası kapsamında çözümlenecek.

Bilim adamı Lord Kelvin 19. yüzyılda: “Bildiğimiz bir konuyu rakamsal değerlerle ifade edebildiğimiz zaman onu gerçekten biliyoruz” demiştir. Bugün de bilinenler sayısal olarak analiz edilebildiğinde, iş zekasını destekleyen somut verilere ulaşılır.

3. VERİ AMBARI

3.1 Veri Ambarının Tarihsel Gelişim Süreci

Donanım yeteneklerinin gelişmesi ve ucuzlamasıyla birlikte kullanılan araçların ucuzlaşıp yaygınlaşması 21. yüzyıla damgasını vuran bir değişim olmuştur. Bu değişimin en büyük etkisi internet ve veri ambarı teknolojilerinde görülmüştür. (Boakye, 1999).

Endüstrideki yenilikleri ve çağdaş yaklaşımları, uygulamak ve benimsemekte atak olan büyük firmalar internet ve veri ambarı fırsatlarını da kaçırmamışlardır. Veri ambarları önceleri firmaların şüpheliğine takılarak hızlı bir yayılım gösterememişlerdir. Hatta kurulumunun karmaşıklığı, belirsiz bir yatırım gerektirmesi ve yüksek maliyetleri nedeniyle tartışma konusu olmuştur (Shin, 2002).

İlk veri ambarları, depolanan veri yığınlarından yararlanmaya yönelik olarak kurulmuştu. Temel amaç, şirketlerin daha iyi yönetilmesini sağlamak, planlama ve kontrol faaliyetlerini güvenilen veriye dayandırmaktı. İlk veri ambarları bunu entegre edilmiş, tarihli, özet verileri işleyerek yapabiliyordu. Ardından daha kapsamlı kararlar alınması için birbirinden farklı verileri içeren kaynaklara ihtiyaç duyulunca veri adacıkları kavramı doğdu. Bu adacıklar veri ambarlarının anahtar bileşeni haline geldi (Zeng vd., 2003). Gelişen mimari seçenekleriyle, veri ambarı uygulama tipleri çeşitlendi ve teknoloji zamanla isteklere göre şekil değiştirebilir hale geldi. Veri ambarları bu haliyle yeni yüzyılın gerektirdiği şekilde şirketlerin esnek ve dinamik bir yapıya ulaşmasında etkin rol oynadı.

Tüm teknolojiler gibi veri ambarı teknolojisi de bir yaşam döngüsü içinde hareket etmektedir. Bundan yaklaşık 15 sene önce, veri ambarları pazarı sadece teknolojiyi uygulayan birkaç vizyoner firmadan oluşmaktaydı (Watson vd., 2002). Uzmanlar, veri ambarlarının henüz hızlı büyüme aşamasında olduğunu ancak kısa vadede olgunluk evresine gireceğini vurgulamaktadır.

İlerleyen yıllarda, şirketlerde veri ambarına duyulan ihtiyaç kolay kolay yok olmayacak gibi görünmektedir. Artan rekabet, müşteri ilişkileri ve kısalan ürün yaşamları gibi parametrelerin hakim olduğu pazarda karar desteği için veri ambarlarına duyulan

ihtiyaç önem kazanarak artacaktır. Sayılan parametrelere sahip olmak isteyen firmalar daha büyük, daha kapsamlı ve daha fazla bilgi çıkarımı odaklı veri ambarları tasarlayacaklardır. Multi-media ve nesne yönelimli veritabanlarının da yaygınlık kazanmasıyla veri ambarı teknolojisi çok boyutlu bir yapıya kavuşacaktır.

3.2 Veri Ambarının Kullanım Amaçları

Operasyonel veritabanı sistemlerinin veri modelleri birim zamanda maksimum işlemin kayıt altına alınmasını sağlamak için optimize edilmiş tasarımlardır. Veri ambarı sistemleri ise şirketlerin verilerini en uygun model ve boyutlarda depolayıp, gerektiğinde hızlı şekilde analiz edilip, raporlanmasına olanak tanıyan tasarımlardır.

İş kullanıcılarının çalıştıkları gün boyunca tekrarladıkları bazı aktiviteler vardır. Bu aktiviteler işletmenin operasyonel sistemlerinde veri oluşturmaktadır. Böylece işletmelerde her gün veriler benzer biçimde yaratılarak devasa bir yığın oluşturulur. Bu veriler IT çalışanlarının zamanlarını harcamakla kalmayıp, operasyonel işlerin aksamasına da yol açar (Kimball ve Ross, 2002). Veri ambarının amaçları iki temel başlıkta ele alınır.

3.2.1 Tek ve Benzersiz Bir Ortam Oluşturmak

Birçok veri adacığının birleşerek oluşturduğu, terabaytlara varan boyutuyla veri ambarları aynı değeri farklı şekillerde ifade eden atomik yapıda veri setlerinin oluşması için uygun bir ortamdır. Tek ve benzersiz bir ortam oluşturmak demek, veri ambarında herkesin doğru olduğunda birleştiği ve kullandığı tek bir olgunun var olması demektir (Roddick vd., 2003).

3.2.2 OLTP Sistemlerinin Yükünü Hafifletmek

Gelişen teknoloji ve artan rekabet dolayısı ile tüm firmalar OLTP sistemlerini kullanmak zorundadırlar. Fakat OLTP sistemi iş kullanıcılarına satışların nasıl gittiği veya onları nasıl geliştirebileceği hakkında bilgi sunamaz (Mattison, 1996). Günlük işlerle zaten yoğun olan operasyonel sistemlerin üzerlerine yüz binlerce satırlık veriyi kapsayan analiz yükü binince iş göremez hale gelmektedirler. Bu durum analitik amaçlar için kullanılacak olan verinin ayrı bir platforma taşınıp offline olarak analiz

edilmesi ihtiyacını doğurur.

Veri ambarlarında şirket bünyesinde bulunan tüm verilerin ya da kullanılması arzu edilen verilerin temiz ve güvenilir bir şekilde toplanması, saklanması ve kullanıma hazır hale getirilmesi sağlanmaktadır. Bu haliyle veri ambarı sistemleri, şirketin iş ihtiyaçları doğrultusunda büyüyen, gelişen ve değişen sistemlerdir. Şirketlere en önemli katkıları, yönetim kademelerinin, ellerindeki verinin doğruluğuna olan güvenini sağlaması ve verilecek olan kararlarda soru işaretlerini ortadan kaldırabilecek altyapıyı sunmasıdır. Veri ambarlarının çıkış noktası da bu ihtiyaçtır. Yöneticilerin, şirketin farklı veri kaynaklarından farklı bilgiler almaları, hangi bilginin doğru olduğuna karar verememelerine, kararlarda gecikmelere ve yanlış kararlar vermelerine neden olmuş ve veri ambarı sistemlerine olan ihtiyacı ortaya çıkarmıştır.

Veri ambarı kullanımı yaygınlaşmadan önce çoğu firmada kullanımda olan yalnızca güncel verilerdi. Geçmişe ait verilere ancak hazırlanması çok uzun zaman alan özel raporlar ile ulaşılabilirdi. Kullanıcılar ve bilgi sistemi uzmanları, verilerin yerlerini tespit etmek ve bunları derlemek için çok fazla zaman harcamak zorunda kalıyorlardı. Veri ambarı, şirket çapında anahtar rol oynayan operasyonel verileri özet, güvenilir ve hızlı ulaşılabilir bir yapıda tutulmasını sağladı (Colin, 1997).

3.3 Veri Ambarının Ayrıntıları

Veri ambarları, merkezi bir yerden yönetilen, çok büyük veri toplama ortamlarıdır. İş değeri, verinin bir yerde depolanmasından çok depolanan verinin kullanılışındaki başarıdan ileri geldiği için, veri ambarları bunu sağlamada çok güçlü teknolojilerdir. Veri ambarları, günümüzde etkin karar destek ve veri analizi için bir önkoşul olarak kabul edilmektedir (Bain vd., 2001).

Veri ambarı sistemlerinde temel mantık verinin uzun süreli ve en detay bazda saklanmasıdır. Performansı artırmak için bilinçsizce yapılacak özet tablo oluşturma, bağımsız data mart oluşturma vs. gibi çözümler, veri ambarı kurulmadan önceki duruma gelinmesine, yine tutarsız bilgilerle karşılaşılmasına ve veri tabanı yönetim maliyetlerinin artmasına neden olacaktır.

Veri ambarları yapısal programlama, ilişkisel veritabanları, bilgisayar destekli tasarım

gereçleri, 4. kuşak programlama dilleri, istemci-sunucu ve nesneye dayalı programlama gibi birçok teknolojiyi bir arada kullanılmasını gerekli kılmaktadır.

Veri ambarları için vurgulanması gereken en önemli nokta ise veri ambarı projelerinin yalnızca bilgi işlem projeleri olmadığıdır. Şirket içinde hem bilgi işlem bölümü hem de iş kullanıcıları diye adlandırabileceğimiz (pazarlama, satış, satın alma, müşteri hizmetleri, muhasebe, üst düzey yönetim vs.) personelin içerisinde bulunması gereken projelerdir. Bilgi işlem bölümü altyapının kurulması ve sistemin hazırlanmasında çalışırken, iş kullanıcısı temsilcilerinin ihtiyaçları düzgün tanımlayıp, sistemin ihtiyaçları karşıladığından emin olmaları ve sistemin kullanılması, kullanıcıların gelişmelerden haberdar edilmesi konularında çalışmaları gerekmektedir (Inmon ve Hackathorn, 1994).

Veri ambarları üzerinde geliştirilen uygulamalar sayesinde, dış ortamdaki değişikliklere uyum süreci kısaltılmıştır. Operasyonel bilgi işlem uygulamalarında amaç bilgisayarların işlem zamanlarını azaltmak iken, günümüzde yoğun rekabet ve çabuk değişen koşullardan dolayı, amaç karar verme zamanlarını kısaltmak olmuştur. Veri ambarı tasarımcıları, operasyonel veri ile stratejik hedefler arasında bir uyum gözeterek veri ambarı tasarımını gerçekleştirmektedir. Geleneksel bilişim sistemlerinin el ile yapılan birçok süreci ortadan kaldırması gibi, veri ambarlarını kullanan karar destek sistemleri de bazı aşamalardaki karar verme ihtiyacını ortadan kaldıracak ve yönetimin bazı katmanları zamanla ortadan kalkacaktır (Kelly, 1996).

Veri ambarlarının doğaları gereği kurumsal yapı üzerinde köklü değişikliklere sebep olacakları kesindir. Kurumsal yapının iyi tasarlanmış bir veri ambarı ile daha hızlı hareket edecek çevik bir yapıya kavuşması da sağlanacaktır (Inmon ve Hackathorn, 1994).

Veri ambarları bilgiyi yararsız gibi görünen yığınların içinden çıkarırken hem iç hem de dış kaynakları kullanırlar ve onların dilini firmanın anlayabileceği dile çevirirler (March ve Hevner, 2005).

Veri ambarlarının tanımı çok çeşitli kaynaklar tarafından yapılmıştır. Bunlar arasında en çok kabul gören ve veri ambarlarını karakteristikleri ile beraber anlatan, veri

ambarlarının babası olarak bilinen Inmon'un (1994) tanımıdır: “Veri ambarları konu odaklı, entegre, zamana duyarlı, değişken olmayan yapıda ve karar desteği için kullanılan sistemlerdir.”

Sınıflandırma: Veriler, kullanıcıların daha kolay yararlanabileceği bir şekilde sınıflandırılır.

Birleştirme: Birçok farklı kaynakta bulunan veriler birleştirilir. Tutarsız ve yanlış veri sistemden temizlenir. Veri tutarlılığı için belli standartlar ve isimlendirme süreçleri oluşturulması şarttır.

Zaman: Veriler anlık durumu değil, bir zaman serisini gösterir. Veri yığınları ne kadar uzun süreli ve derinlikli bir zaman serisini kapsıyorsa karar desteği o kadar sağlıklı olur.

Statik: Daha önce de anlatıldığı gibi veri ambarları üzerlerindeki sorgular çevrim dışı olarak işletilir. Bu yüzden veri ambarları periyodik olarak genişletilir. Hem veri işleme hem de analiz etme yeteneğine sahip olan bir sistemin kaynak tüketimi verimsiz olacaktır.

3.4 Veri Ambarında Kullanılan Veri Türleri

Veri ambarı sistemi içindeki verilerin çeşitleri mevcuttur. Bunlar üç temel başlık altında incelenebilir: içsel veri, dışsal veri ve metadata.

3.4.1 İçsel Veri

Şirketlerin günlük işlemleri sonucu oluşturduğu, OLTP sistemlerinden veri ambarına aktarılan her veri içsel veri olarak adlandırılır. Firmaların kendi bünyelerinde oluşan ürünlerle, hizmetlerle, müşterilerle veya çalışanlarla ilgili olan ve analitik sürece tabi tutulabilecek her veri bu kapsamda ele alınabilir. Bunlar tek bir formata dönüştürülmeden önce bölümlerin kendi operasyonel veritabanlarına işlenir. Sonrasında, offline olarak inceleme yapılması için veri ambarına gönderilir (Turban vd., 2006).

3.4.2 Dışsal Veri

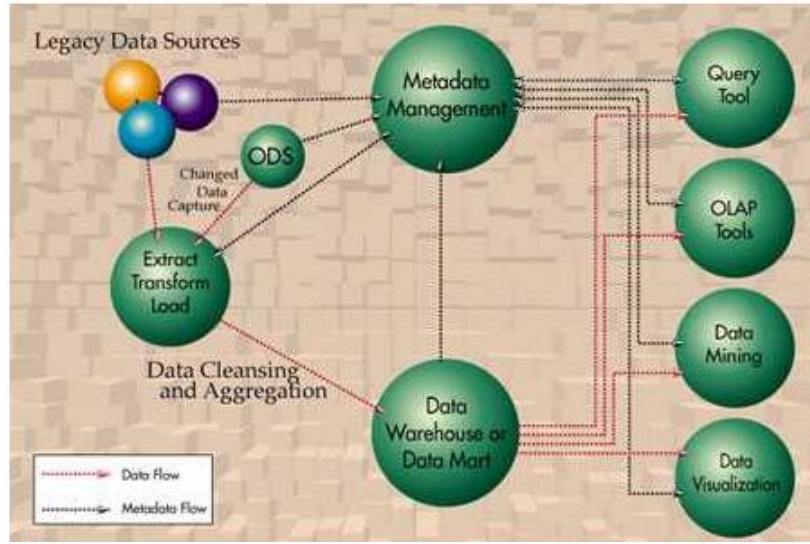
Günümüzde kapalı sistemlerin varlıklarını sürdüremedikleri açık bir gerçektir. Sistemler ancak çevreleriyle olan ilişkilerini devam ettirdikleri sürece uzun bir yaşama kavuşabilirler. Her ne kadar şirketlerin motoru içsel aktiviteler olsa da çevreden kaynaklanan bazı etki ve değerlere göre planlama yapmak durumundadırlar. Veri ambarları da benzer şekilde, kimi analizleri yaparken dışsal verilerin oluşturduğu kısıtları ya da verileri göz önünde bulundurmamak zorundadır. Kimi zaman kısıt oluşturan bu parametreler meteorolojik veriler, halka açık ortamlarda bulabileceğimiz kanunlar, uydu verileri, taşınabilir veri depolama aygıtlarından gelen veriler, kayıtlar diye uzayıp gitmektedir.

Veri ambarlarının kullanıldığı bazı sektörlerde dışsal verinin önemi fazlaca artmaktadır. Örnek olarak sağlık sektöründe sadece bir kurumun tüm sağlık bilgi tabanını oluşturamayacağı düşünülürse tedavi yöntemleri, sağlık standartları, yerel demografi ve sağlık istatistikleri büyük önem kazanmaktadır. Bu durum, sağlık sektörünün çevresel bağımlılığının çok olmasından kaynaklanır ve benzer bir çok sektör bu kümeye dahil edilebilir (Roddick vd., 2003).

3.4.3 Metadata

Metadata isminden anlaşıldığı gibi bir veri tipi değil, veri hakkındaki bilgidir. Veri ambarının bileşenlerini, verilerini ve süreçlerini kapsar. Metadata deposu veri ambarının yöneticisine de kullanıcıya da işleri açısından büyük destek sağlar. Öyle ki sadece veri ambarındakileri tanımlamakla kalmaz, üzerinde yapılan tüm tasarım değişikliklerinin, bileşenler ve mimariye yapılan tüm müdahalelerin de bir tarihçesini saklar (Vassiliadis, 2000).

Veri ambarı işleyişi için tanımlanması gereken metadata her süreç için ayrı bir yapıda oluşturulur. Daha sonra ayrıntılı olarak irdelenecek olsa da örneğin kaynak sistemlerden verinin çekilmesini sağlamak ya da iş alanına has yapılan özel yazılım geliştirmeleri için bir metadata oluşturulabilir (Kimball ve Ross, 2002).



Şekil 3.1 Metadata yönetiminin grafiksel gösterimi [6]

Şekil 3.1'de görüldüğü gibi veri ambarında analiz için kullanılması gereken veriye her ulaşmak istendiğinde, metadata kullanılacak olan verinin hangisi olduğunu belirler ve deposuna yerleştirir. Daha sonra bu bilgi kaynak veriyi çıkartma, dönüştürme ve veri ambarına yönlendirme ETL işlemlerini tanımlar (Almeida vd., 1999). Sistemin dinamosu gibi çalışan metadataya tüm veri ambarı işlevleri bağımlıdır. Çünkü metadata tablo ve boyut isimlerini, veri birleştirme ve hesaplama yöntemlerini, biçim ayarları ile verinin soyutlanmasına kadar her bilgiyi içerir (Mattison, 1996). En çok kullanılan iki metadata türü aşağıda açıklanmıştır.

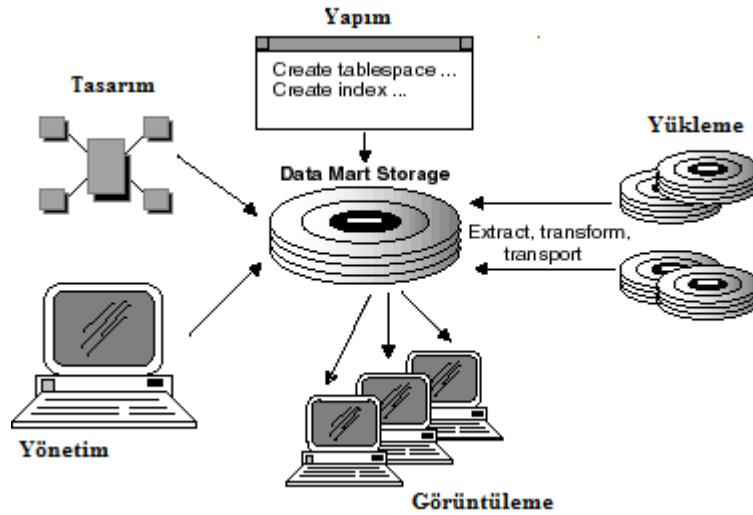
Operasyon seviyesinde: Asıl veri hakkındaki bilgi operasyon seviyesindeki metadatada barındırılır. Verinin veritabanı içinde nasıl konumlandırıldığını, operasyonel sistemlerdeki yapısını; çıkarım, dönüştürme ve yükleme yöntemlerini içeren, veri ambarı yöneticilerine yönelik bir metadatadır (Gray ve Watson, 1998). Operasyon seviyesinde metadata başka bir deyişle verinin sicilini (ne, nerede, nasıl vs.) açıklayan bir nüfus kimliği gibidir.

İş seviyesinde: İş seviyesinde metadata son kullıcılara yöneliktir. Veri ambarı yöneticilerine yönelik olmayan bu metadata iş terimlerini ve tanımlarının, verinin kime ait olduğunu ve veri hakkında kime başvurulması gerektiğini anlatır (Chaudhuri ve Dayal, 1997).

3.5 Data Mart

Data Mart terimi, küçük veri ambarı anlamında kullanılarak, bir kaç departmanın veri gereksinimlerini karşılamak üzere tasarlanmış veri yapıları olarak tanımlanır (Inmon ve Hackathorn, 1994).

Data martlar, veri ambarlarının bir alt kümesidir. Aslında data martlar veri ambarları ile benzer özelliklere sahiptir. Ancak veri ambarları, varlık gösterdiği iş dünyasındaki tüm verilere farklı açıdan bütünleşik bir bakış sağlarken, data martlar bu bütünleşik ve karmaşık yapı içinden sadece belirli bir konu ile ilgili olan kısmı, örneğin belli bir bölümün kullanacağı verileri kapsar. İşletme organizasyonunda bazı seviyelerdeki karar vericilerin, işletmeye ait tüm veriler üzerinde yapılacak analizlere ihtiyacı olmayabilir. Bu kişiler sadece kendi birimleri ile ilgili verilere ulaşarak bunlar üzerinden analiz yapmayı ve buna göre kararlarını yönlendirmeyi isteyebilirler. Bu kişilerin, veri ambarındaki tüm karmaşıklık ile uğraşmasına gerek kalmadan, sadece ilgilendikleri veriler üzerinde analizleri gerçekleştiren bir sisteme ihtiyaçları vardır. Bu amaçla, veri ambarı sistemi üzerindeki tüm verilerin sadece belli bir kısmından oluşan alt bir sistem, data mart olarak hazırlanır. Şekil 3.2’de data martların veri ambarı sistemi içindeki rolü gösterilmiştir.



Şekil 3.2 Data mart [7]

3.5.1 Veri Ambarı ile Data Martlar Arasındaki Farklar

Veri ambarları ile data martlar arasında iki önemli fark bulunmaktadır. Data martlar belirli bir ihtiyacı karşılamak üzere oluşturulurlar. Veri ambarları ise, belirli bir ihtiyacı giderme yerine, kurumsal bir veri deposu olarak hizmet verir. Veri ambarları ile data martlar arasındaki bir diğer önemli fark ise data marttaki verilerin veri ambarındaki verilerden daha tanecikli bir yapıda olmasıdır.

Veri ambarları kurumsal strateji doğrultusunda tasarlanırlar. Data martlar ise taktik planlar çerçevesinde ve daha özel gereklilikler için uygulanırlar (Bain vd., 2001).

3.5.2 Data Martlara İhtiyaç Duyulan Durumlar

Data martların oluşturulma sebeplerini aşağıdaki gibi maddeler halinde sıralayabiliriz;

- Sorgu hızlarını, üzerinde arama işlemi yapılacak verilerin hacmini azaltarak artırmak için,
- Verilerin yapılarını, kullanıcıların verilere erişim için kullandıkları araçlara uygun olan forma sokmak için,
- Verileri belirli bir erişim kontrol stratejisi çerçevesinde bölümlemek için,
- Verileri farklı donanım platformlarına bölmek için (Anahory ve Murray, 1997).

Kurumsal ihtiyaçları data martlar ile çözmeye çalışmak, projenin maliyetlerini ve başarısını orta vadede negatif etkilemektedir. Data martlar genel olarak çeşitli fonksiyon ya da konu başlıkları için yapılandırıldıklarından, çoğu kez merkezi bir veri ambarı altyapısı gibi veri bütünlüğünü sağlayamamaktadır.

Data mart tasarımına başlamadan önce, data mart stratejisinin kurumun yapısına uygun olup olmadığına karar verilmesi gerekir. Data mart stratejisinin uygun olup olmadığını bulabilmek için aşağıdaki aşamalar takip edilebilir (Anahory ve Murray, 1997).

- Organizasyon içinde doğal bir fonksiyonel ayrım olup olmadığı,
- Organizasyon içindeki veride doğal bir ayrım olup olmadığı,
- Kullanılan ya da teklif edilen kullanıcı erişim araçlarının kendi veri yapılarının olup olmadığı,
- Herhangi bir alt yapı mimarisinin data mart uygulamasını gerekli kılıp kılmadığı araştırılır.

4. VERİ AMBARI TASARIMI

Bir veri ambarı tasarlanmadan önce tüm proje sürecinde geçerli olacak stratejilerin belirlenmesi gerekir. Bu stratejiler aşağıdaki temel bileşenleri kapsamalıdır:

Kural: Güvenlik yetkileri ve bakım sıklığı gibi veri ambarının kullanımını ve idaresini düzenleyen kuralları içerir.

Dönüşüm: Kaynak sistemlerden transfer edilecek veri, silme ve dönüşüm işlemleri uygulanarak veri ambarı sistemine uyumlu bir forma getirilmek zorundadır.

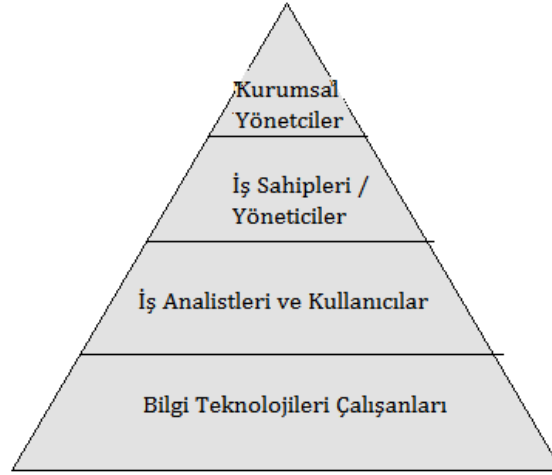
Depolama: Dönüştürülerek veri ambarında depolanan veriler; esnek, ulaşılabilir ve yönetilebilir olmalıdır (Bain vd., 2001).

4.1 Analiz Süreci

Bir veri ambarı projesi müşteri gereksinimlerinin toplanması ile başlar. İyi sonuçlar elde etmek için uygun görüşme planı takip edilmelidir. Bu planla, organizasyondaki yöneticilerin beklentilerini öğrenmek ve onlardan projeye pozitif katkısı bulunacak insanları tesbit etmek için fikir almak mümkün olacaktır.

Mevcut iş sistemini ve kurallarını öğrenmek ve yararlı verileri elde etmek için, doğru insanlara doğru sorular yöneltilmelidir. Böylece veri ambarının boyutsal modelinde yer alacak ölçümler ve boyutlar kolaylıkla ve uygun bir şekilde belirlenebilir. Örneğin bir müşteri yöneticisi ya da iş analisti müşterilerin toplam satışlarının bölge, satıcı, müşteri vb. yönleriyle ilgili bilgilerle ilgilendiklerini söyleyebilirler. Ayrıca stokların sipariş sayısı ile olan ilişkisinin onların işleri için önemli olduğunu iletebilirler. Böylece ölçümler ve boyutlar uygun şekilde seçilmiş olur.

Ölçümler ve boyutların belirlenmesi süresince bir hata yapılırsa bu projede en başa dönülmesi gerektiği anlamına gelir. Bu nedenle kararların dikkatli bir şekilde belirlenmesi gerekir.



Şekil 4.1 Veri ambarı analiz etki piramidi (Stackowiak vd., 2007)

İş Gereksinimleri: İş gereksinimlerini kararlaştırmak; veri ambarının veya veri paketlerinin hedeflerini tanımlamak, projeye veya müşteriye göre gerekli diğer konuların kapsamalarını çizmek anlamına gelir. İş gereksinimlerini iyi tanımlamanın bir sonucu olarak, veri ambarının özellikleri aşağıdaki terimlerle tanımlanabilir (Bain vd., 2001):

Konu Alanları: Farklı iş birimlerinin ilgi alanları farklı olacaktır. Örnek olarak, pazarlama bölümünün ilgisini çekebilecek konular şunlardır: Pazar araştırması, rekabet analizi, alıcı davranışı, pazar bölümleri, ürün (pazar uyumu), fiyatlandırma ve bütçelendirme kararları, ürün kararları, promosyon kararları, ürünlerin tedarik yöntemlerinin kararları ve yönelimlerini tahmin etme.

Örnek Ürün: Özel konu alanlarıyla sonuçlanacak önemli bir basamaktır. Pazarlama bölümü örneğinde olduğu gibi, siparişler, promosyonlar, pazarlar, satışlar, ve zaman döngüsü gibi spesifik konu alanları belirlenebilir.

Ayrıntı: Taneciklenme ne kadar artar ise, detay miktarı o kadar artacaktır. Boyutlardaki seviye sayısının artışıyla taneciklenme sağlanmaktadır.

Boyutlar: Analiz gereksinimlerine dayanır, aşağıdaki boyutlar genel kabul görmüştür:

- Takvim hiyerarşisi ile zaman (gün, hafta, ay, çeyrek, yıl)
- Müşteri grupları (müşteri, pazar bölümü, pazar, endüstri)
- Ürün aileleri (ürün, ürün ailesi, ürün zinciri)
- Coğrafya ve yer (satış temsilcisi, bölge, semt, ülke, uluslar arası bölge)

- Organizasyon yapısı (bölüm, iş ünitesi, şube, stratejik iş ünitesi, şirket)

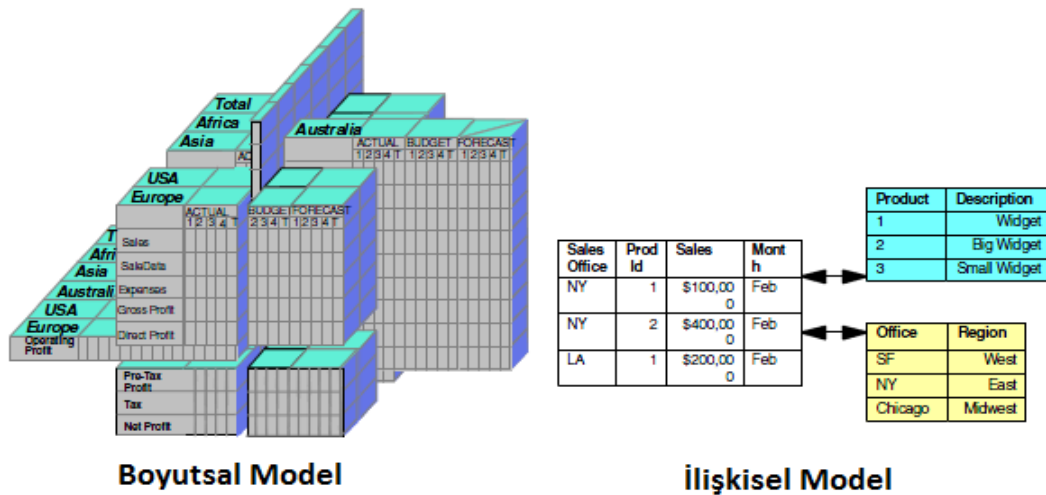
İş Hedefleri: İş hedeflerinin iki özelliği vardır: geniş hedefler ve spesifik hedefler. Müşteri servisini geliştirmek geniş hedeflere örnek olarak verilebilir. Bunun yanında verilen periyotta veri ambarında ürünlerin stok seviyesini belirlemek spesifik hedeflere örnek olabilir.

Veri Kaynakları: Yararlı veri kaynaklarını kararlaştırmak, veri toplamak ve veritabanından veri ambarına taşımak için gerekli olanları ayırmak amacıyla uygun planlar hazırlamak çok önemlidir.

Son Kullanıcı Gereksinimleri: Veri ambarı sisteminin müşterisi son kullanıcı olacağından, sistemin onların gereksinimlerine göre tasarlanması çok önemlidir. Son kullanıcılar sorguları ve raporları hazırlamayı, veri ambarını görüntülemeyi, yaptıkları modelleri görmeyi ve veri analizlerini hiçbir zorlukla karşılaşmadan kolayca yapmayı isterler. Kısacası, kullanıcıların onayladığı ve kullanımını kolay buldukları GUI'leri tasarlamak çok önemlidir.

4.2 Veri Modelleme Teknikleri

Bir veri tabanı yaratırken iş sürecinin özelliklerini ve aralarındaki ilişkileri gösteren bir veri modeli kurmak ilk adım olmalıdır. Şekil 4.2'de fiziksel veri modelleme teknikleri görülebilir.



Şekil 4.2 Fiziksel veri modelleri (Reinschmidt ve Francoise, 2000)

4.2.1 Varlık İlişkili Modeller

Varlık ilişkili veri modelleme tekniği iki ana kavram üzerine kuruludur: varlıklar ve bu varlıklar arasındaki ilişkiler. ER modelleri özel ilgi alanları için veri modeli üretiminde kullanılır.

Gerçek dünya nesnelerini temsil eden varlıklar, özelliklerine ve karakteristiklerine göre sınıflandırılabilirler ve gözlemlenebilirler. Detaylı ER modelinde, varlığın tek bir tanımlayıcısının olması kritiktir. Bu tanımlayıcı, aday anahtar olarak adlandırılır ve birincil anahtar olarak değinilen anahtarı seçmek için kullanılır.

Eğer bir öznitelik, tablodaki kayıdın benzersiz bir tanımlayıcısı ise ve boş değilse, birincil anahtar olarak kullanılabilir. Ayrıca birden fazla öznitelik bir araya gelebilir ve birlikte bir anahtarı oluşturabilirler. Bu anahtar türü birleşik birincil anahtar olarak isimlendirilir.

Yabancı anahtarlar farklı tablolarda temsil edilen iki varlık arasında ilişki kurmak amacı ile kullanılır. Bu durumda iki anahtar da aynı karaktere sahip olmalıdır.

Varlıkların karakteristik nitelikleri öznitelikleri olarak verilmiştir. Mantıksal modeldeki bir varlık ve bir öznitelik fiziksel modelde sırasıyla veritabanı tablosu ve tablo sütunuyla eşleşir.

Varlık ilişkili modelleme tekniği veri ambarı modellemesi için kullanılabilir, fakat bu kullanım bazı dezavantajlar doğurur. Son kullanıcılar açısından bu modeller anlaşılabilir, yönlendirilemez ve kolayca unutulabilir olur. Veri ambarlarının amacı olan sezgisel ve etkili olarak veriye erişebilme ER modelleme tekniği ile mümkün olmaz.

4.2.2 Boyutsal Model

Boyutsal modelleme veriyi standart bir çerçeve içinde sunmaya çalışan mantıksal bir veri ambarı modelleme tekniğidir. Veri iş birimine göre tanımlanan ölçümlerle görselleştirilmiştir. Bu sayısal veriyi (değerler, tekrarlar, ağırlıklar, dengeler) ve veritabanı tablolarının içeriğindeki oluşumları, özetlendirme ve verinin tekrar düzenlenmesini tutan modeldir. Bunun yanında veri analizini desteklemek amacıyla verinin sunumunu da içermektedir.

Veri ambarında, boyutsal modelleme ER modellemesine göre daha basit ve daha kolay

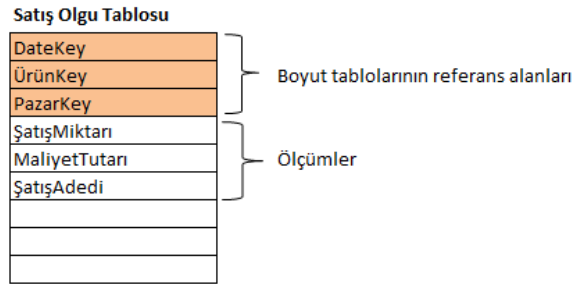
anlaşılabilir olmasına rağmen daha pahalıdır. ER modellemesinden farklı olarak, boyutsal modelleme iş dünyasındaki kullanıcıların gereksinimlerini karşılayabilme bakımından çok güçlüdür (Kimball ve Ross, 2002). Bir boyutsal model iki temel kavramdan oluşur: Olgular ve Boyutlar.

4.2.2.1 Olgu Tabloları

Olgu tabloları verileri, genelde sayısal ve eklentili, bir başka deyişle analiz edilebilir ve incelenebilir olarak (satış, maliyet, kar gibi) temsil eder. Olgu tablolarında iki çeşit alan vardır. Olgu tablosunda bilgi içeren alanlar ölçüm olarak adlandırılır. Olgu tablolarındaki ikinci tür alanlar ise, boyut tablolarına bağlantının kurulduğu referans alanlardır (Bain vd., 2001).

Bir olgu tablosu detaylı olguları ya da yığınlaştırılmış olguları içerir. Yığınlaştırılmış olguları içeren olgu tabloları genellikle özet tablolar olarak adlandırılır. Birçok olgu tablosu eklenebilir. Ayrıca yarı-eklenebilir ya da eklenemez de olabilirler. Eklenebilir olgular basit aritmetik toplamayla yığınlaştırılabilirler. Bunlara yaygın örnek ise satışlardır. Eklenemez olgular bir bütün olarak toplanamaz. Bunlara örnek ise ortalamalardır. Yarı eklenebilen olgular ise bazı boyutlar boyunca yığınlaştırılırken bazılarında ise bu mümkün değildir. Bunlara örnek ise herhangi birisinin basitçe bakarak hangi ortalama seviyesinde olduğunu söyleyemeyeceği envanter düzeyleridir.

Bir olgu tablosu her bir boyut tablosu için bir referans alanı tutar. Bu referans alanlarda boş değerler bulunamaz. Olgu tablosunda, kayıtların tekliği tablonun referans alanları tarafından garanti altına alınır. Eğer durum böyle değil ise her bir kaydın tekliğini sağlamak için tablo içine bir alan eklenmelidir. Şekil 4.3’de bir olgu tablosunun yapısı görülebilir.



Şekil 4.3 Olgu tablosunun yapısı (Reinschmidt ve Francoise, 2000)

Olgu tablosundaki mevcut kayıtlar, bir hata düzeltilmediği veya bir proje değişimi yapılmadığı takdirde asla güncelleştirilmez. Tabloya yeni kayıtlar eklenebilir. Arşivleme dışında mevcut kayıtlar kesinlikle silinmezler.

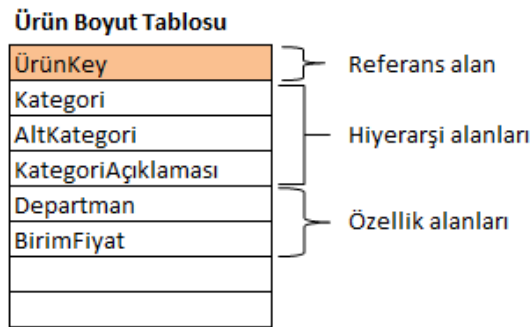
4.2.2.2 Boyut Tabloları

Boyut kullanıcıların iş sorularını yanıtlamayı mümkün kılan, veriyi sınıflandıran bir yapıdır. Yaygın kullanılan boyutlar müşteriler, ürünler ve zamandır. Bir boyut, olgu tablosundaki değerlere istenilen şartlara göre erişebilmeyi sağlayan bilgileri bulundurur.

Boyut bir iş ölçüsüdür. Ölçütler birçok farklı şekilde gösterilebilir. Örneğin zaman boyutu, saat, hafta, gün, yıl ve çeyrek gibi alt boyutlara ayrılabilir. Boyutlar, kendi içinde değişik ölçü düzeyleri şeklinde gruplanabilir. Bu değişik düzeylere hiyerarşiler denir. OLAP sistemleri, boyutları ve bunlardan oluşturulan hiyerarşileri kullanarak trend analizleri yapabilir, veri dilimlenebilir, detaylı alt veriye ulaşılabilir, farklı dönemler için karşılaştırmalar yapılabilir.

Bir olgu tablosundaki olguların tek başlarına çok az anlamları vardır. Boyut tabloları, o bilgilerin daha anlaşılır ve yararlı bilgiler olarak kullanılmalarını sağlar. Boyut tabloları, olgu tablolarından daha küçüktür. Olgu tabloları yıldız şemasının toplam hacminin %98'inden fazlasını oluştururlar.

Olgu tablolarına daha fazla kayıt eklenirken, boyut tabloları üzerinde çok büyük değişiklikler yapılmaz. Boyut tablolarına sadece yeni bir mağaza açıldığında, yeni bir müşteri geldiğinde ya da yeni bir ürün satılmaya başlandığında kayıt eklenecektir. Olgu tablosu ise her gün devam eden satışlardan dolayı sürekli büyüyecektir. Şekil 4.4'te bir boyut tablosunun yapısı görülebilir.



Şekil 4.4 Boyut tablosunun yapısı (Reinschmidt ve Francoise, 2000)

4.2.3 Veri Ambarı Şemaları

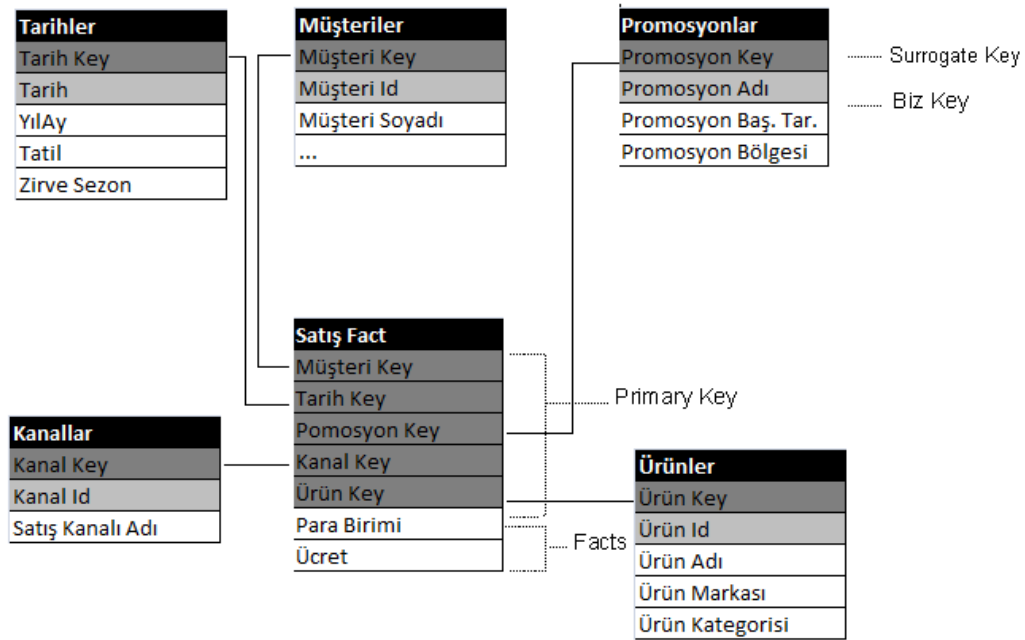
Veri ambarı şemaları kaynak veri ve kullanıcının ihtiyaçlarına göre kurgulanır. İki tip veritabanı şeması vardır. Bunlar yıldız şeması ve kar tanesi şemasıdır.

4.2.3.1 Yıldız Şeması

Yıldız şeması iyi tanımlanmış eklenti yolları ve birkaç tablo ilişkisinin basit yapısı olarak tanımlanabilir. Bu veritabanı dizaynı, operasyonel veritabanının kullanımındaki normalleştirilmiş yapıya oranla, performansı yüksek ve basit bir şema sunar.

Yıldız şeması çevresi boyutsal veri ile sarılmış gerçek datayı depolayan fiziksel veritabanı yapısıdır. Yapılacak analizin karmaşıklığına göre boyut tabloları bir kaç taneden başlayarak artacaktır. Bu şema verinin sadece okunan veri olarak, referans verinin ise zaman içinde değişecek veri olarak kullanmasında etkilidir (Anahory ve Murray, 1997).

Yıldız şeması boyut tablolarının kırılıp normalize tablo olmaması için denormalize edilmiştir. Bu nedenle son kullanıcı görüşüne yatkındır. Yıldız şemasını Şekil 4.5'te görebilirsiniz.



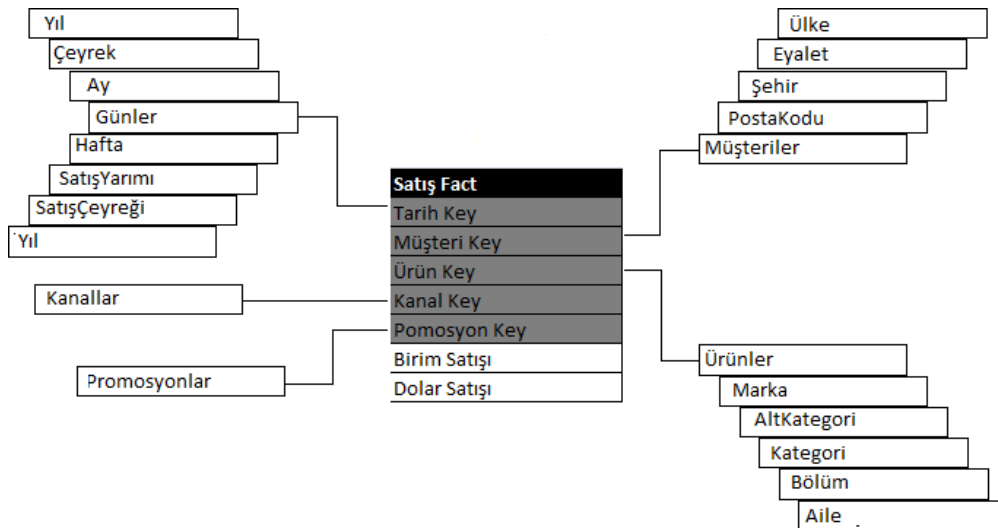
Şekil 4.5 Satış analizi için yıldız şeması örneği (Bain vd, 2001)

4.2.3.2 Kar Tanesi Şeması

Kar tanesi şeması temel olarak yıldız şeması gibi olsa da yıldız şemasından daha karmaşık bir veri ambarı modelidir. Şeklinden dolayı kar tanesi ismini almıştır. Fazlalıkları elimine etmek için normalize edilmiştir.

Boyut tabloları büyük bir tablo yerine birden çok tabloya gruplanmıştır. Mesela yıldız şemadaki bir ürün boyut tablosu kar tanesi şemasında ürün, ürün kategori tablosu ve ürün üretici tablosuna normalize edilebilir. Bu yerden kazanım sağlarken, boyut tablosunu arttırıp daha fazla yabancı bağlantıya ihtiyaç duyar. Sonuç olarak daha karmaşık bir tepki sürecine ve tepki performansında azalmaya yol açar. Kar tanesi şeması Şekil 4.6'da görülebilir.

Yıldız şeması, boyutsal model olarak işletme varlıklarının ve onlar arasındaki ilişkilerin gösterilmesinin en basit yoludur. Bununla beraber bazen boyut tablolarından bir kaç tanesi çok hızlı bir şekilde büyüyüp, çok büyük bir tablo halini alabilir. Böyle durumlara karşı veri tabanı tasarımcıları ve geliştiricileri tarafından kar tanesi şeması yaratılmıştır.



Şekil 4.6 Satış analizi için kar tanesi şeması örneği (Bain vd., 2001)

4.3 Veri Yakalama, Dönüştürme ve Yükleme

Veri ambarı tasarımının bir diğer bölümü data dönüştürme ve yükleme stratejisinin dizyan işlemidir. Veri ambarına data yükleme işlemi birkaç aşamadan oluşur. Bu aşamalar: veri yakalama, veri dönüştürme ve veri yüklemedir (Bain vd., 2001).

4.3.1 Veri Yakalama

Yakalama işlemindeki veri kaynağı, dosya biçimlerini, ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemlerini içermektedir. Verinin hangi türde bir dosyadan yakalandığı, veri yakalamada kullanılan tekniğe bağlıdır. Bu teknikler: kaynaktan veri çekme, VTYS log yakalama, tetiklemeli yakalama, uygulama destekli yakalama, zaman bağımlı yakalama ve karşılaştırmalı yakalama teknikleridir.

Kaynaktan veri çekme, kaynak verinin belli bir zamandaki statik görüntüsünü verir. Bu teknik, sürekli bir tarihçeye ihtiyaç duyulmadığında kullanılabilir. Örneğin, veri bloğunun tamamı alındığında ve depodaki mevcut veri aynı kaynaktan alınan farklı zamana ait yenisiyle değiştirildiğinde iyi sonuçlar vermektedir. Veri ambarı yıllık olarak güncellenir ve önceki yıla ait eski veri, yıl sonunda arşive aktarılır.

VTYS log yakalama, verinin VTYS günlük sisteminden yakalanmasını sağlar. Bu teknik, sadece ilgilenilen verinin yakalanabilmesini sağlamak üzere, günlük kayıtların nasıl bir biçimde tutulduğunun bilinmesini gerektirir.

Tetiklemeli yakalama, tetikleyiciler ve saklı prosedürler, ekleme, güncelleme ve silme senaryoları gibi önceden tanımlı olaylar gerçekleştiğinde çalışmak üzere hazırlanmakta ve pek çok veritabanı yönetim sistemi tarafından desteklenmektedir. Tetikleyici, belirlenen olayın gerçekleştiğini farkeder ve prosedürü aktif hale getirir. Prosedürün kontrol ve bakımı, kullanıcının sorumluluğundadır. Tetikleyici, veritabanı yönetim sisteminin değil prosedürü yazanın denetiminde olduğundan değişikliklere de açıktır.

Uygulama destekli yakalama, operasyonel kaynaklardan veri toplamak için program yazımını içeren tekniktir. Bu teknik, test ve bakım sorumluluklarını içerecek şekilde, tamamen programcının denetimi altındadır.

Zaman bağımlı yakalama, kaydın son yakalamadan sonra değişip değişmediğini belirlemek üzere kullanılan bir zaman belirtecine dayanan bir tekniktir. Eğer bir kayıt değişmişse ya da yeni bir kayıt eklenmişse, bu kayıt işlenmek üzere başka bir dosya veya tabloya aktarılır.

Dosya karşılaştırma, veri kaynağının belirli bir zamanda alınmış kopyası, bir dosyaya kaydedilir ve sonra daha önceki bir halini içeren dosya ile karşılaştırılır. Belirlenen değişiklikler ya da eklemeler, daha sonra işlenmek üzere ayrı dosyalara kaydedilir.

Tüm bu tekniklerin yukarıda anlatıldığı gibi bazı avantajları vardır ve planlayıcının en iyi sonucu alabilmesi için tüm bu tekniklerin bir kombinasyonunu kullanması yerinde olacaktır (Bain vd., 2001).

4.3.2 Veri Dönüştürme

Dönüştürme işlemi, yakalanan kaynak veriyi veri ambarına yüklenebileceği uygun biçim ve yapıya dönüştürür. Dönüştürme işlemi, kaynak verideki bozuklukları ortadan kaldırarak hedef veri ambarı için yüksek kaliteli verinin elde edilmesini sağlar. Temel dönüştürme teknikleri; yapısal dönüştürme, içerik dönüştürme ve işlevsel dönüştürme 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.

4.3.3 Veri Yükleme

Uygulamaya koyma işlemi, dönüştürme işlemi ile yaratılan dosyaları işleyerek, bunları ilgili veri ambarı ya da veri pazarına yüklemektir. Bu işlem için 4 temel teknik kullanılır:

Yükleme: Hedef veri ambarı tablolarını, dönüştürme işleminde yaratılan tablolarla değiştirir. Eğer veri ambarında bu tablolar yoksa, tablolar yaratılır.

Ekleme: Varolan tabloların sonuna dönüştürme sonucunda ortaya çıkan tablodaki yeni veriler eklenir.

Yapıcı Birleştirme: Var olan hedef tablolara yeni verileri ekleyerek durumu değişen kayıttaki bir zaman değerini günceller.

Yıkıcı Birleştirme: Yeni verileri var olan kayıtların üzerine yazar.

5. ÇEVİRİMİÇİ ANALİTİK SÜREÇ

5.1 OLAP Kavramının Tarihi

OLAP teknolojisinin temelleri 1961 yılında Charles Bachman tarafından ilk veri tabanı yönetim sisteminin geliştirilmesi ile atılmıştır. OLAP'ın temeli olan çok boyutlu analiz, 1962 yılında Kenneth Iverson tarafından yazılan “A Programming language” adlı kitabın yayınlanmasıyla ortaya çıkmıştır ve ilk olarak IBM tarafından 1960'lı yılların sonlarında kullanılmaya başlanmıştır. 1969 yılında Dr.Codd ilişkisel veri tabanı yapısını bulmuştur [8].

OLAP terimi ilk olarak 1993 yılında Dr.Codd'un hazırladığı bir makalede kullanılmıştır. Makalede OLAP'ın tanımıyla birlikte bir ürünü OLAP olarak tanımlayabilmek için gerekli olan 12 kural belirlemiştir. 1994 yılında Nigel Pendse ve Ricard Greeth, kurumların OLAP uygulamalarını satın alabilmeleri ve kullanabilmeleri için bağımsız bir kaynak olan OLAP ile ilgili bilgileri www.olapreport.com'da* yayınlamaya başlamışlardır [9].

Pendse ve Creeth, OLAP kurallarının çok uzun ve akılda kalmasının zor olduğunu düşünüp OLAP'ı yeniden tanımladılar. 1994 yılında ilk ROLAP aracı MicroStrategy DSS Agent ürünü, 1995 yılında ilk HOLAP aracı Holos 4.0 piyasaya çıkmıştır (Nagabhushana, 2006).

5.2 OLAP Teknolojisinin Özellikleri

OLAP uygulamaları aşağıda yer alan temel özellikleri sağlamalıdır [10]:

5.2.1 Veri Küplerine Çok Boyutlu Yaklaşım

Analizler sırasında kullanılan her türlü kırılıma boyut adı verilir. Örneğin demografik veriler (yaş, cinsiyet, eğitim durumu), sayısal veriler, işlem miktarları, gerçekleşen ve bütçelenen değerler, ürün tipleri, ürün özellikleri ve zaman boyut olarak kullanılabilir. Yöneticiler ve analistler, çalışmaları sırasında, tüm bu tanımlanan verileri yatay veya

* Bu web sitesi daha sonra isim değiştirerek <http://www.bi-verdict.com/> sitesi olmuştur.

düşey eksenlerde konumlandırarak görmek isterler. İlişkisel veri tabanları, bu esneklikteki raporlara cevap veremezler. Eksiklikler raporlama araçlarının yetenekleri ile belirli bir noktaya kadar tolere edilebilir. Fakat işin içine daha karmaşık analizler girdiğinde, bu raporları bir OLAP yapısı kurgulamadan oluşturmak imkansız hale gelir. Bu yapıda raporları hazırlamak için ilişkisel veri tabanları üzerinde karmaşık SQL kodları yazmak ya da raporlama aracının sahip olduğu programlama dilinin özellikleri üzerinde uzun süre uğraşmak gerekir. Bu durum analizi yapan kişileri, işin özünden uzaklaştırarak, analiz gerektirebilecek verilerden başka konularla zaman kaybetmelerine neden olur. Teknik olanaklarla ilgili araştırma yapmaya başlayıp sonuç alamadıkları durumlarda daha kolay şekilde ulaşılabilecek veriler tercih edilir. Böylesi bir seçim yanlış analizlere yol açar. Bu nedenle, iş zekası programının pratik olmasının yanında, fazla teknik bilgi kullanmadan raporların alınabilir olması çok önemlidir.

OLAP yapısında kullanılan boyutların başka bir özelliği de bunlar üzerinde hiyerarşiler tanımlanabilmesidir. Hiyerarşiler sayesinde, hem toplamlara ulaşmak kolaylaşmakta, hem de farklı gruplar için, farklı senaryolar hazırlayabilme esnekliği oluşmaktadır.

5.2.2 Karmaşık Hesaplamalar

OLAP sistemlerinin gerçek performansı karmaşık hesaplamaları yapabilme güçleri ile ölçülmektedir. Analiz yapanlar için rakamlardan çok, yüzdesel dağılımlar önemlidir. Birkaç yıllık satışlar içinde binlerce ürün türü için gün bazlı satışları yüzdesel olarak analiz edip sıraya dizebilmek RDBMS ile saatler sürecektir bir raporun çalışmasını gerektirebilir. Oysa uygun bir OLAP sistemi ile bir günlük satışlar ve birkaç yıllık satış rakamlarının raporlanması arasında bir fark olmayacaktır. Satış tahminlerinde genellikle hareketli ortalama ve yüzdesel artış gibi trend analizleri kullanılır. Kullanılacak OLAP yapısında bu şekildeki hesaplamalara izin verir bir yapının bulunması gerekir.

5.2.3 Zaman Boyutu

Zaman boyutu, neredeyse her analizin temel bileşenidir. Zaman, diğer boyutlardan farklı olarak, kendine has bir sıralama yapısı içerir. Alfabetik (Ocak her zaman Şubat'tan önce gelmelidir) veya nümerik sıralamalardan (12/31, 01/01'den önce

gelmelidir) her zaman farklıdır. OLAP sistemleri, zamanın bu şekillerde sıralanabilmesine imkan verirler.

5.3 OLAP Kuralları

Bir veri yapısının OLAP olarak nitelendirilebilmesi için 12 kuralın gerektiği önceki bölümlerde belirtilmişti. Bu kurallar sırası ile şunlardır [11]:

1. Çok Boyutlu Kavramsal Bakış

OLAP araçları veriyi çok boyutlu bir şekilde tutabilmeli, kullanıcılar bu veriye kolay ve hızlı şekilde erişebilmeli ve analiz yapabilmelidirler. Ayrıca kesme ve veri küpünün döndürülmesi gibi işlemlerin zaman ve çaba harcamadan yapılması mümkün olmalıdır.

2. Saydamlık

OLAP farklı veri kaynaklarını desteklemeli, herkese açık ve anlaşılabilir bir yapısı olmalıdır. Kullanıcının verinin hangi kaynaklardan geldiğini bilmesine gerek olmadan veriye eksiksiz bir biçimde erişmesi gerekir.

3. Erişilebilirlik

OLAP araçları heterojen veri kaynakları ve OLAP kullanıcı arayüzü arasında bulunmaktadır. OLAP araçları farklı kaynaklardan gelen fiziksel verileri saklamak, bu verilere erişebilmek ve herhangi bir dönüşüm yapabilmek için kendi mantıksal şemalarını oluşturmalarıdır.

4. Tutarlı Raporlama Performansı

Model içindeki boyutların sayısı ya da verinin hacmi, OLAP aracının performansını etkilememelidir. Raporlama performansı boyut sayısı ya da verinin hacmi ne olursa olsun sabit kalmalıdır.

5. İstemci Sunucu Mimarisi

Çevrimiçi analitik işlem yapılacak olan büyük miktarlardaki veriler genelde ana bilgisayarda depolanmakta ve bu verilere kişisel bilgisayarlar üzerinden erişilmektedir. Bu nedenle OLAP araçları sorunsuz bir şekilde istemci/sunucu ortamında çalışabilmeli ve istemci/sunucu mimarisinin tüm özelliklerine sahip olmalıdırlar.

6. Genel Boyutlandırılabilirlik

Tüm boyutlar sahip oldukları özellikler açısından birbirine denk olmalıdır. Seçilen boyutlara ek işlev özellikleri verilebilmelidir. Temel veri yapıları, formüller ve raporlama biçimleri, herhangi bir veri boyutuna özel olmamalıdır.

7. Dinamik Seyrek Matris Kullanma

OLAP araçları, performans azalmasını önlemek için, fiziksel depolama seviyesinde büyük ve seyrek matrislerin üstesinden etkili bir biçimde gelebilmelidir.

8. Çoklu Kullanıcı Desteği

Bazen aynı anda, birden fazla analist aynı OLAP modeli ile çalışmak ve aynı veri üzerinde farklı modeller oluşturmak ihtiyacı duymaktadır. OLAP araçları, çoklu kullanıcı erişimi, bütünlüğü ve verinin güvenliğini sağlamalıdır.

9. Sınırsız Çapraz Boyutsal İşlemler

OLAP araçları kullanıcıların farklı veri boyutları karşısında hiç bir kısıtlama ve engelleme olmadan çoklu hesaplamalar ve formüller tanımlamasına izin vermelidir.

10. Sezgisel Veri İşleme

Satır ve sütunlardaki verinin ayrıntılandırılması, birleştirme ve diğer veri düzenlemeleri uygun şekilde gerçekleştirilmeli ve kullanımı kolay bir arayüz olmalıdır.

11. Esnek Raporlama

Mümkün olan her yönde raporlama yapılmasını sağlamalı, boyutları ihtiyaç duyulan şekilde rapora yerleştirmeyi mümkün kılmalıdır. Bu nedenle OLAP araçları, satırlarda, kolonlarda ya da sayfa başlıklarında 0'dan n'ye kadar tüm boyutları gösterebilmelidir.

12. Sınırsız Boyutlar ve Seviyeler

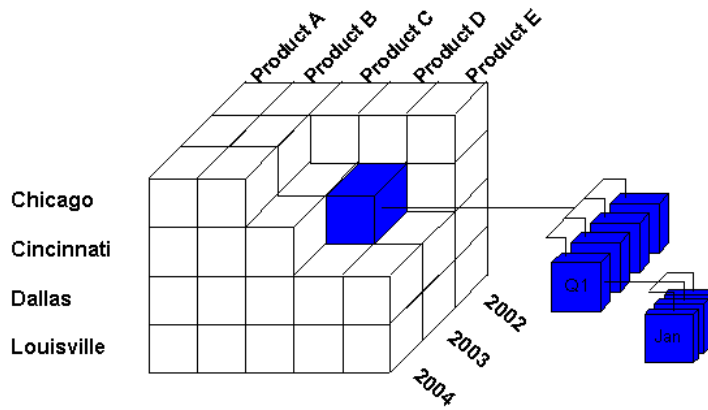
Analitik bir model içerisinde istenilen sayıda boyut bulundurulabilmelidir. Bu boyutlardan her biri kullanıcı tarafından tanımlanan sınırsız sayıda grup seviyesini kabul edebilmelidir.

5.4 Temel OLAP Kavramları

5.4.1 Hiyerarşi

Hiyerarşiler, boyutu oluşturan seviyelerin sırasını belirlemektedir. Düşük seviyeli kavramlardan yüksek seviyeli kavramlara sıralama yaparlar. Hiyerarşiler sayesinde kullanıcılar verileri önce özet seviyede inceleyip daha sonra ayrıntı seviyelerine inebilmektedir. Örneğin ilk olarak tüm yıla ait satış verilerini inceleyip daha sonra istedikleri seviyeye inerek veriyi inceleyebilmektedirler (Kramer ve Nisbet, 2008).

Hiyerarşiler kullanıcıların sayısal değerler içinde hareket ederek eldeki veriyi farklı açılardan görmelerini sağlar. Örnek vermek gerekirse, kullanıcı bir ürün ailesinin alt ürün türlerine bakabilir. Sonrasında bölgesel satış detaylarını inceleyebilir. Bu satışların sadece 2010 yılının ilk çeyreğine ait olanlarını görüntüleyebilir. Sonuçta tüm bu verilerin toplamına yani toplam satış tutarına göz atabilir. Kullanıcı verinin içinde hareket ederek karar vermek için gerekli bilgilere kolayca ulaşabilir.

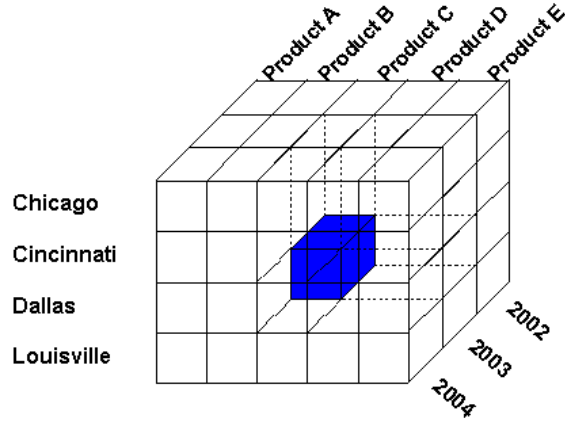


Şekil 5.1 Zaman hiyerarşisi [12]

5.4.2 Ölçüm

Ölçüm analiz konusunun niceliğini ifade etmektedir. Örnek olarak satış verileri alınacak olursa önemli ölçümler kar, maliyet, birimler, indirimler vb. olacaktır. Bu şekilde boyutlar içindeki karmaşık ham veri OLAP sayesinde tüm kullanıcılar tarafından anlaşılabilir olmaktadır. OLAP kar, maliyet gibi ayrılmış ölçümlerin gereken karmaşık modeller ile işlenmelerine imkan tanımaktadır. Bir ölçüm, işletmenin

performansının değerlendirilmesinde kullanılmak üzere hesaplanan belirli bir konunun toplam değeridir (Koutsoukis vd., 1999).



Şekil 5.2 Bir ürünün belirli bir tarihteki satış ölçümü [13]

5.4.3 Seviye

Hiyerarşinin içindeki her bir unsur seviyedir ve boyutlar seviyelerden oluşmaktadır. Örneğin zaman boyutundaki yıl-mevsim-ay-gün hiyerarşisinin içindeki yıl, mevsim, ay ve gün birer seviyedir. Boyutların bazen tek seviyesi olabilmektedir. Seviyeler üyelere oluşmaktadır.

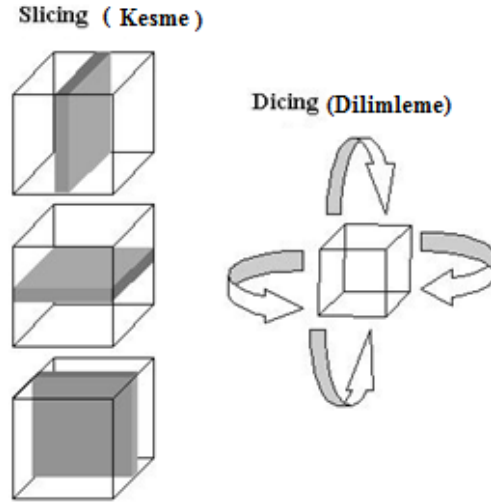
5.4.4 Üye

Boyutu oluşturan seviyelerin her bir parçasına üye denilmektedir. Üye boyutların verileridir. Her seviye bir ya da birden fazla üyeye sahip olabilir. Örneğin zaman boyutunun, yıl seviyesinin üyeleri 2006, 2007 ve 2008 olabilir (Kramer ve Nisbet, 2008).

5.4.5 Kesme ve Dilimleme İşlemleri

Kesme, boyutların bir ya da daha fazla üyesi için tek bir değerle ilgili olarak düzenlenmiş küpün bir alt kümesidir. Kesme işlemi tek bir boyutun seçimine ve küpün belirli bir bölümü üzerinde odaklanılmasına dayanmaktadır. Örnek olarak zaman, bölge, ürün tipi veya bu boyutların herhangi bir kombinasyonunu vermek mümkündür [15].

Dilimleme işlemi ise iki veya daha fazla boyut üzerinde seçme işlemi gerçekleştirilerek alt küpler oluşturulması işlemidir (Han ve Kamber, 2006).

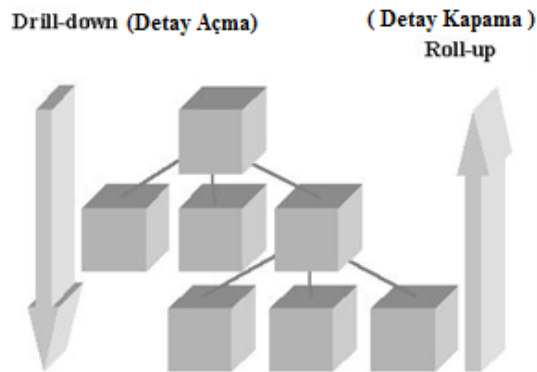


Şekil 5.3 Kesme ve dilimleme işlemleri [14]

5.4.6 Detay Açma ve Detay Kapatma İşlemleri

Detay açma işlemi bir ya da daha fazla boyut hiyerarşisinde toplam seviyesinden detay seviyesine inilerek verilerin incelenmesi anlamına gelmektedir (Datta ve Thomas, 1999).

Detay kapatma işleminde ise küp verisi üzerinde kümeleme işlemi gerçekleştirilmektedir. Detay açma işleminin aksine detay seviyesinden toplam seviyesine çıkılarak veriler incelenmektedir.



Şekil 5.4 Detay açma ve detay kapatma işlemleri [14]

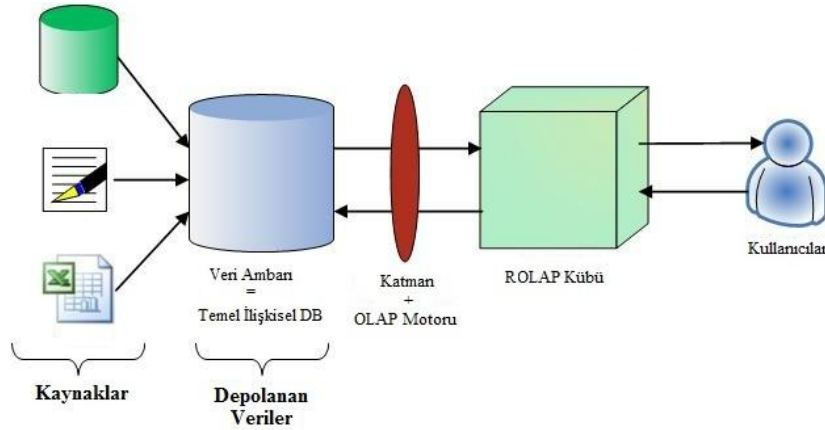
5.4.7 Pivot İşlemi

Küp verisinin eksenleri etrafında döndürülerek veriye farklı açılardan bakılmasını sağlayan bir işlemdir (Han ve Kamber, 2006). İki boyutlu veri yapıları üzerinde de pivot işlemini gerçekleştirmek mümkündür.

5.5 OLAP Depolama Yöntemleri

Olgular ve bilgilendirici tablolar içeren bir yıldız şeması kullanarak, bir veri ambarı yaratıldığında, sonraki adımda mutlaka OLAP servislerince kullanılacak olan, boyutlar ve ölçümleri içeren küpler kurulmalıdır. Bu küpler ROLAP, HOLAP veya MOLAP kullanarak saklanabilir. Farklı durumlar, farklı yükleme metotlarının kullanılmasını gerektirir.

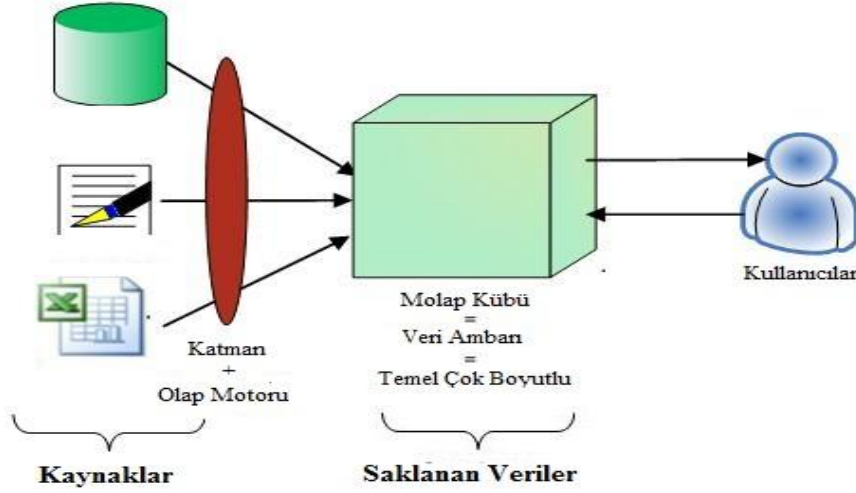
ROLAP, bir ROLAP yükleme şeması kullanırken, OLAP servisleri, kümeleri ilişkisel veri ambarı içerisine yerleştirir. Küpler ve kümeler kurmak için veri ambarındaki veriler kullanılır. Bu şu anlama gelir; özel çok boyutlu veri yapıları, veri toplamak için kullanılmaz. ROLAP yükleme şemasıyla birlikte çalışan bir OLAP aracı, tüm OLAP fonksiyonlarını desteklemelidir ve kümelere izin vermelidir. Şekil 5.5 bir ROLAP çözümünü yansıtmaktadır.



Şekil 5.5 Rolap çözümü [16]

Eğer ROLAP kullanılıyorsa, OLAP veri tabanının OLTP veri tabanı performansını azaltıp azaltmadığını test edilmeden, veri ambarı OLTP veri tabanı içine yerleştirilmemelidir. OLTP ve OLAP'ı bir veri tabanı içine kombine ederek kullanılan ROLAP, geri dönülmez bir biçimde OLTP sistemini etkileyecektir.

ROLAP her şeyi ilişkisel veri tabanı içinde bırakırken, MOLAP bunun tam aksini yapar. MOLAP, OLAP servisi için gerek duyulan tüm veriyi çok boyutlu farklı bir veri tabanı içine yerleştirir. Bir MOLAP çözümü Şekil 5.6'da gösterilen yapıya benzer bir özellik taşır.



Şekil 5.6 Molap çözümü [16]

Farklı küplerdeki özdeş boyutlar için aynı bilgi tablosu kullanıldığında, aynı veri, bu bilgi tablosunu kullanan tüm küpler için çok boyutlu veri tabanı içine kopyalanır. Eğer 5 küpünüz varsa ve hepsi bir zaman boyutu yapmak için aynı zaman bilgi tablosunu kullanıyorsa, çok boyutlu veri tabanı içinde 5 kopya özdeş zaman bilgisine sahip olunur. Bu durum veri tabanında fazla yer kullanmanın dışında bilgiyi veri tabanına aktarırken extra zaman kaybına sebep olur. Eğer veri ambarı içindeki her bir bilgi tablosu, çok boyutlu veri tabanı içine sadece bir kere transfer edilirse, o zaman bu tek kopya tüm küpler tarafından kullanılır ve bu bilgi tablosunun her bir küp için özdeş boyutlar içine defalarca kopyalanmasından daha az zaman alır.

5.5.1 ROLAP ile MOLAP Arasındaki Farklar

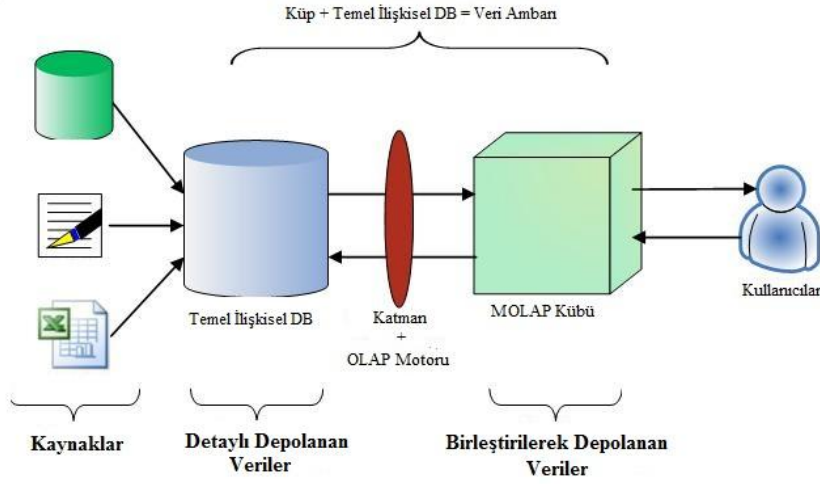
ROLAP kullanırken eldeki veri, ilişkisel tablolar içine yüklenir. Bu tablolar, ilişkisel veri tabanı formatındadır ve görüntülenebilir. ROLAP'ta veri tekrarlama yoktur, çünkü veri ambarındaki veri direkt olarak OLTP sistemi tarafından kullanılır. ROLAP'ın kullandığı disk alanı daha azdır.

MOLAP, ekstra disk alanına ihtiyaç duyar. Çünkü veri ambarındaki bilginin, OLAP servislerinin veri üzerindeki sorgulamaları gerçekleştirmeleri için, çok boyutlu veri tabanı içine taşınması gerekir. Ama bu veri sıkıştırılmıştır, bitmap indeksleri kullanır ve orijinal ilişkisel veri ambarından daha az yükleme alanına ihtiyaç duyar.

MOLAP, oldukça hızlı ve verimlidir. Veriyi yüklemek için çok boyutlu veri tabanı kullanır. Verinin dönüş zamanı tutarlı ve hızlıdır. MOLAP, veri ambarındaki veriyi çok boyutlu veri tabanı içine transfer etmek için, oldukça geniş bir zamana ihtiyaç duyar. Bu zaman saatlerden günlere kadar sürebilir.

Özet olarak, ROLAP daha az yükleme alanına, daha uzun sorgulama zamanına ihtiyaç duyar ve ilişkisel data martlar kullanır. ROLAP, nadiren ulaşılan veya çok geniş veri içeren veri ambarları için idealdir. Birçok kuruluş, geçmiş verilerini iki gruba ayırır, bunlar mevcut tarihsel veri ve arşivlenmiş tarihsel veri olarak adlandırılır. Mevcut tarihsel veri, sıklıkla kullanılan ve belli bir zamanı genellikle 2 yılı kapsayan veridir. Arşivlenmiş tarihsel veri ise çok nadiren kullanılır ve birinci gruptaki zaman dilimini aşan zamanlar için kullanılır. Arşivlenmiş tarihsel verinin boyutu birçok yılı kapsadığı için çok geniş olabilir. Mevcut tarihsel veri ve veri ambarlarının çoğu için en iyi cevap MOLAP'tır.

Çok fazla yükleme alanı kullanmayan veya tüm verinin çok boyutlu bir veri tabanına transfer edilmesini gerektirmeyen ama ROLAP'tan daha iyi performans gösteren bir sistem istenebilir. Bu durumlar için HOLAP formu kullanılır. HOLAP, ROLAP ile MOLAP arasındaki orta bir yerdir. Veriyi, veri ambarından çok boyutlu veri tabanına taşımaz. Bu açıdan HOLAP, ROLAP gibidir. Buna karşın, kümeler çok boyutlu veri tabanı içine yüklenir. Şekil 5.7 bir HOLAP çözümünü göstermektedir.



Şekil 5.7 Holap çözümü [16]

OLAP servisleri içinde, hesaplanmış bileşenler özet veriler için kullanılır. Hesaplanmış bileşenler hafızaya yüklenir ve ihtiyaç duyulduklarında tekrar hesaplanırlar. Yüksek seviyede kümelerle sahip olan bir veri tabanı içinde, hesaplanmış üyeler, OLAP servisi tarafından çok hızlı bir şekilde bulunabilir. Çok kümeli bir HOLAP veri tabanı içinde, hesaplanmış üyelerin değerlerini bulmak çok hızlıdır, çünkü çok boyutlu veri tabanı içine sadece kümeler yüklenmiştir. Hesaplanmış üyeler için HOLAP veri tabanı MOLAP veritabanından daha hızlı olabilir. ROLAP gibi, HOLAP'ın da detay sunması yavaştır. Ama kümelerden bilgi getirişi hızlıdır. HOLAP hesaplanmış üyeler dışında, ROLAP'tan daha iyi performans gösterir, fakat MOLAP'tan daha yavaştır. Eğer mevcut tarihsel veri iki gruba ayrılırsa, daha eski olan, sıklıkla erişim gerektirmediği takdirde, HOLAP olarak yüklenebilir. Bu kararlar şirketin ihtiyaçlarına ve veri ambarlarının isteklerine bağlıdır. Tek bir çözüm yoktur.

5.6 OLAP ve OLTP Karşılaştırılması

OLTP sistemleri sonuçları mümkün olduğunca çabuk dağıtmak zorundadır. Bir sipariş işlendiğinde, hem müşteri hem de siparişi alan değerli bir zaman harcamaktadır. Analitik sorgular OLTP sistemlerini fark edilir bir ölçüde yavaşlatırlar. Çünkü bunlar, işlem zamanını ve hafıza kaynaklarını kullanmaktadırlar. Hem çevrim içi işlemlerin hem de analitik soruların aynı zamanda çalışması, aralarındaki kayıt kilitleme özelliği tarafından yavaşlayabilir.

OLTP için kullanılan veritabanları genellikle, ayrı kayıtların tekrar ele geçirilmesi ve güncellenmesi için bir veri tabanını optimize eden süreci normalleştirir. Normalleştirilmiş veri tabanı içinde, her bir parça bilgi sadece bir kere depolanır dolayısıyla tek bir yer içinde güncelleştirilir. OLAP içinde kullanılan analitik sorular, veri tabanı şeması çoklu kayıtları sorgulamaya göre tasarlandığında çok daha hızlı çalışır. Bu tasarım stratejisi, boyutsal modelleme olarak adlandırılır ve sonuç bir yıldız şeması gibi ortaya çıkar.

OLTP sistemleri, tablolar sınırlı sayıda hedeflenmiş indekslere sahip olduklarında, daha verimli çalışırlar. Her bir indeks, her veri değişiminde, değiştirilmelidir. Çok fazla indeksin olması, OLTP'nin normal kayıt güncelleme işlemini yavaşlatır. Birçok indeks olduğunda OLAP sistemi çok daha iyi çalışır. Veri analiz edildiğinden, güncelleştirme yapılmaz, dolayısıyla geniş sayıdaki indekslerin güncelleştirilmesiyle ilgili zaman harcama problemi ortadan kalkar.

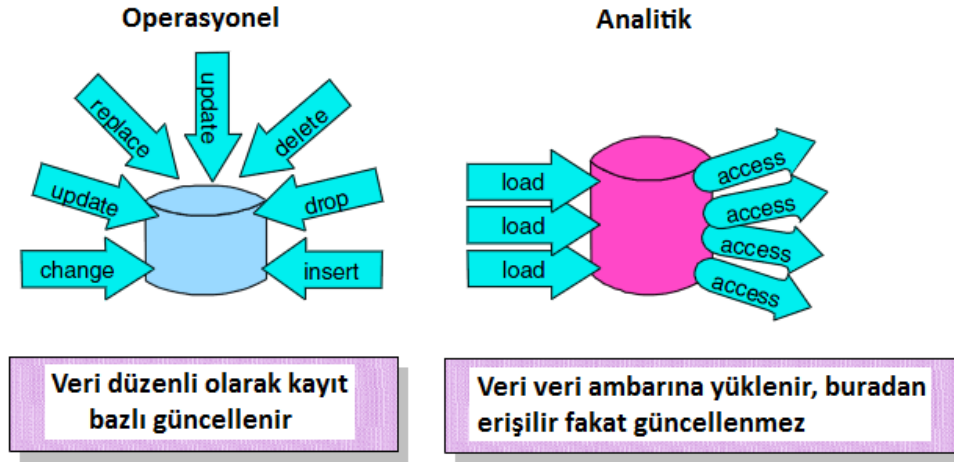
Bir OLTP sistemi içinde, özet değerler hesaplanmak istenir. Çünkü ayrı kayıtlar değişir ve her bir kayıt değiştiğinde tüm özet değerlerin değiştirilmesi zorunlu hale gelir. Bir OLAP sistemi içinde, toplanmış değerlerin analiste hemen sunulabilmesi için, birçok özetleme istenir. Analistin görüntülemek için hangi faktörlerin kombinasyonunu seçtiği önemli değildir.

Bir OLTP sistemi için en uygun veri depolama alanı, ilişkisel veri tabanı sistemidir. Bu tip veri tabanı sistemi gerçekte, özel olarak OLTP sistemlerinin amaçlarını karşılamak için geliştirilmiştir. OLAP sistemi için kullanılan veri, bir ilişkisel veri tabanı sistemi içinde toplanabilir. Fakat aynı zamanda çok boyutlu veri yapılarının içinde de toplanarak performans kazanma ihtimali vardır. Çok boyutlu veri tabanı sistemleri, OLAP tarafından kullanılan, analitik sorgulamaya ait özellikli ihtiyaçları karşılamak üzere yaratılmışlardır.

Veri işlemenin amaçları, OLTP ve OLAP için birbirinden oldukça farklıdır. Hem OLTP hem de OLAP sorgulamaları için aynı veri tabanını kullanmak mümkündür fakat her ikisi için tek bir veri tabanı optimize etmek imkansızdır. Bu iki tip sorgulamayı optimize etmek için farklı stratejiler vardır ve bu stratejiler birbirine zıttır. Eğer OLTP

ve OLAP sisteminizin her ikisinden de mümkün olan en iyi performansını almak istiyorsanız, onları ayrı veritabanlarına bölmek gerekmektedir.

Şekil 5.8’de operasyonel (OLTP) ve analitik (OLAP) veri tabanlarının temel farkları görselleştirilmiştir.



Şekil 5.8 Operasyonel ve analitik veri tabanlarının farkları
(Reinschmidt ve Francoise, 2000)

6. HİZMET SEKTÖRÜNDE İŞ ZEKASI UYGULAMASI

Sağlık sektörü, insan yaşamındaki kritik yeri nedeni ile teknolojik gelişmelerin yakından takip edildiği ve uygulandığı bir sektördür. Bu ihtiyaç göz önünde bulundurularak misyonunu; “Tüm insanların kişisel sağlık durumunu iyileştirerek ve genel refahı geliştirerek, müşfik bakım ve yenilik destekli uzman tıbbi hizmet sunarak, ekip çalışması ve sürekli gelişmeyle hastalar ve müşteriler için değer üreterek, tıbbi eğitim ve araştırmada liderlik sergileyerek, her bireye saygıyla yaklaşarak, hizmet verdiği insanların sağlık durumunu iyileştirmeyi taahhüt eden bir bölgesel sağlık bakım sistemidir.” olarak tanımlayan bir sağlık kurumu için iş zekası uygulaması yapılmıştır.

Uygulamaya geçmeden önce sağlık sektöründe iş zekası ile çözüme kavuşabilecek güncel sorunlar üzerinde durulmuştur. Elde edilen bilgiler ile projenin amaç ve kapsamı belirlenmiş ve uygulama aşamasına geçilmiştir.

6.1 Sağlık Sektöründe İş Zekası ile Çözümlenebilecek Güncel Sorunlar

Kısıtlı kaynakların etkin kullanılabilmesi için hastanelerin finansal olarak başarılı olmaları ve performans düzeylerinin artırılması sağlanabilir. Her alanda olduğu gibi sağlık alanında da suistimaller oldukça yaygındır ve çoğu kamu finansmanına dayalı hizmetlerde olduğu için ülke ekonomisi üzerinde oluşturduğu yük çok büyüktür. Ayrıca, fatura yolsuzluğu, yeşil kart suistimali gibi suistimallerin tespiti çok zordur. Ülkemizde şu anda şikayete dayalı bir tespit sistemi vardır. Bu sistem ile suistimallerin önüne geçmek mümkün değildir. İş zekası çözümleri ile bu süreçler sağlıklı bir şekilde kontrol edilebilir, veri madenciliği işlevi ile sahtecilik tespitleri üzerine başarılı çalışmalar yapılabilir.

Hizmet maliyetlerinin belirlenmesi sağlık hizmetini sunan ve satın alanlar açısından oldukça önemlidir. Sağlık hizmetleri sunan kurumlar maliyetlerini denetim altına alabilmek için maliyetlerine etki eden faktörleri bilmeli ve maliyetlerini minimize edici çözümler üretmelidirler. İş zekası çözümlerini kullanan bir kurumda veri ambarından alınacak bilgiler ile bu faktörler kolaylıkla belirlenebilmektedir.

Sağlık hizmetlerinde kalite birçok faktörden etkilenen ve hastanın algılama düzeyine göre değişkenlik gösteren bir kavramdır. Bu nedenle kalite göstergelerini tespit ederken, hastalık grupları, hastanın demografik özellikleri, hastanın sigorta durumu,

klirik ve hizmet kalitesi gibi deęişkenlerin bir arada düşünülmesi ve deęerlendirilmesi gerekir. İş zekası çözümleri kullanıcıların bu farklı bilgilere aynı ortamdan kolayca ulaşabilmelerini sağlamaktadır.

Ortalama yaşam süresinin artışıyla beraber, kronik hastalıkların görülme sıklığı ve buna paralel olarak getirdiğı mali yük giderek artan bir seyir izlemektedir. Bu noktadan hareketle, kronik hastalıkların ortaya çıkmasını engelleyecek proaktif çözümler geliştirilmesi gerekmektedir. Veri ambarından elde edilecek kronolojik verilerden çıkarılacak istatistiksel sonuçlar bu çözümlere ışık tutacaktır.

Saęlık hizmetlerinin sunumunda yanlış teşhis ve tedavi tespiti güç sorun alanlarından birini teşkil etmektedir. Hastanın problemlerinin teşhis ve tedavisinde hekime yardımcı olacak veri ambarı yaklaşımlı bir veri bankası oluşturulmasına ve teşhis veya tedavi sırasında hekime veri, çözüm, risk ve önerileri otomasyona baęlı olarak sunabilecek sistemlere ihtiyaç duyulmaktadır.

Saęlık sektöründe iş zekası çözümlerinin faydalı olacağı dięer konular şunlar olacaktır:

- Saęlık personelinin performansının izlenmesi
- Hasta akış planlarının yapılması
- Tıbbi tedavi süreçlerinin optimizasyonu
- İlaç kullanım hata ve yan etkileri için erken uyarı sistemi
- Veri madenciliğine dayalı olarak hasta ve ilaç kullanımının profilendirilmesi
- Türkiye ilaç kullanım haritasının hazırlanması
- Kronik hastalıklarda veri madenciliğine dayalı olarak ilaç kullanım alışkanlıkları ve risk tespitinin yapılması
- İlaçların birim maliyetlerinin hesaplanması
- İlaçların inovasyon maliyetlerinin belirlenmesi
- Bioterörizme karşı saęlık veritabanı oluşturulması
- Afet telafisinde önceliklerin ve minimum maliyetlerin belirlenmesi

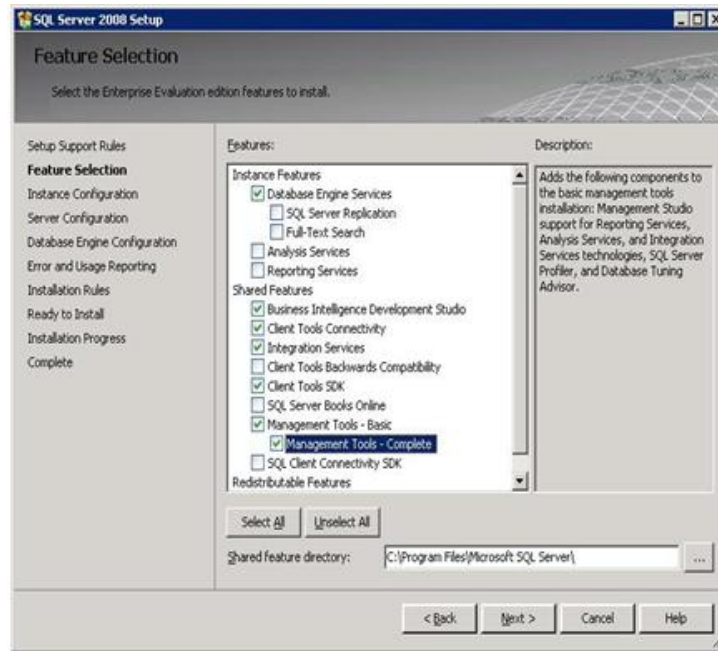
Yukarıda ifade edilen sorunların çözümü için saęlık sektöründe iş zekası uygulamalarının kullanılması zorunludur.

Kurumda yapılan analizler sonucu strateji haritası şu tanım üzerine kurulmuştur. Strateji haritası çoęu saęlık bakım sektörü haritası gibi, kuruluşun misyon ve

vizyonunun net bir ifadesi ile başlayarak kuruluşun nihayi hedefleriyle, daha maddi olan büyümeye ve verimliliğe yönelik arzulanan finansal sonuçlar arasında doğrudan bir nişan çizgisi oluşmasını sağlar. (Kurum strateji haritasına Ek2'den ulaşılabilir.)

6.2 Business Intelligence Development Studio

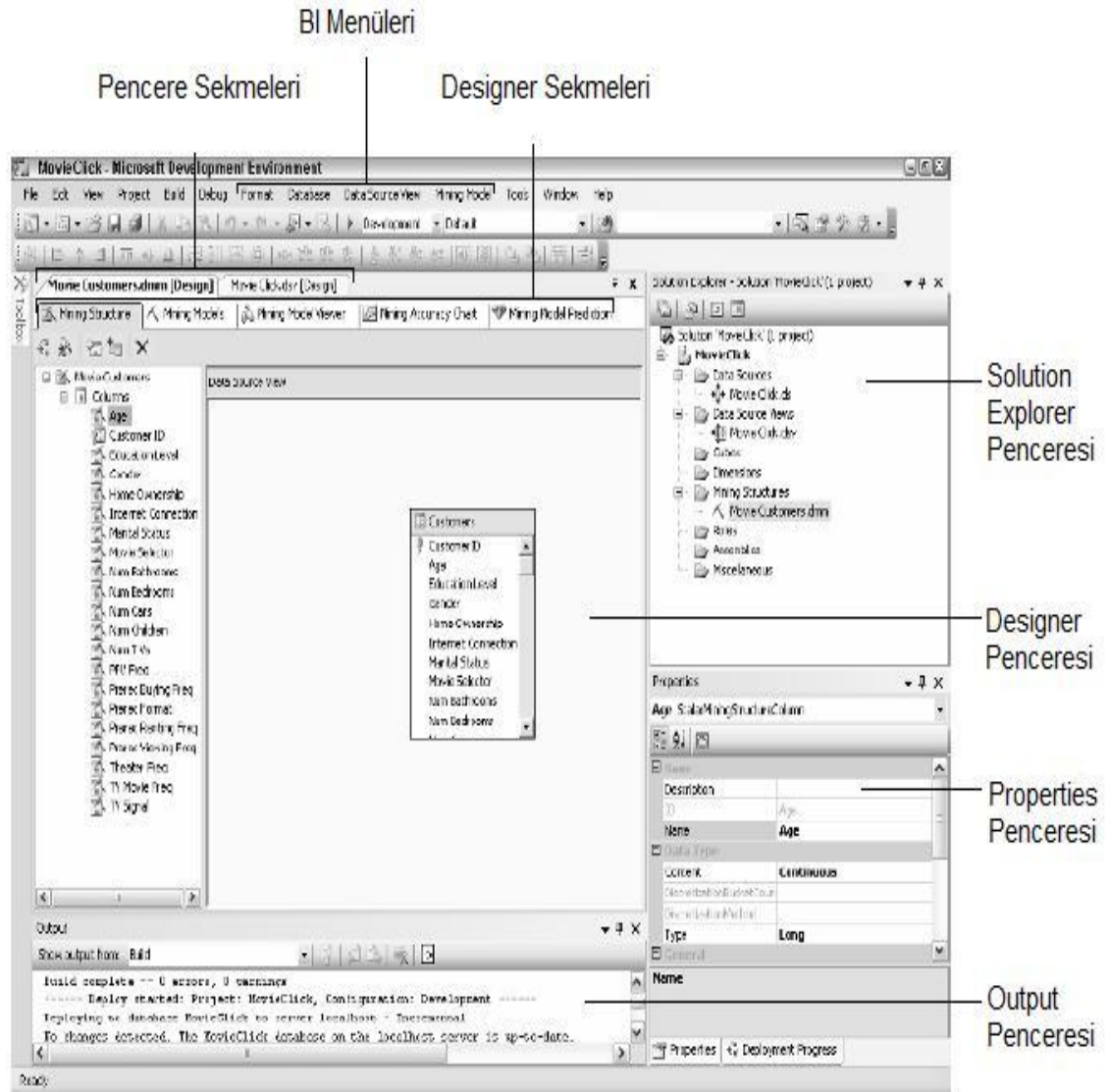
Microsoft SQL Server'ın araçlarından biri olan Business Intelligence Development Studio (BIDS); iş zekası projelerinin gerçekleştirildiği özel bir visual studio ortamıdır. MS SQL Server 2008'in kurulumunu yaparken, Şekil 6.1'deki gibi Visual Studio 2008 versiyonunun özel bir sürümü olan BIDS'nun da yüklenecek şekilde işaretlenmiş olması gerekir.



Şekil 6.1 Business intelligence development studio kurulumu

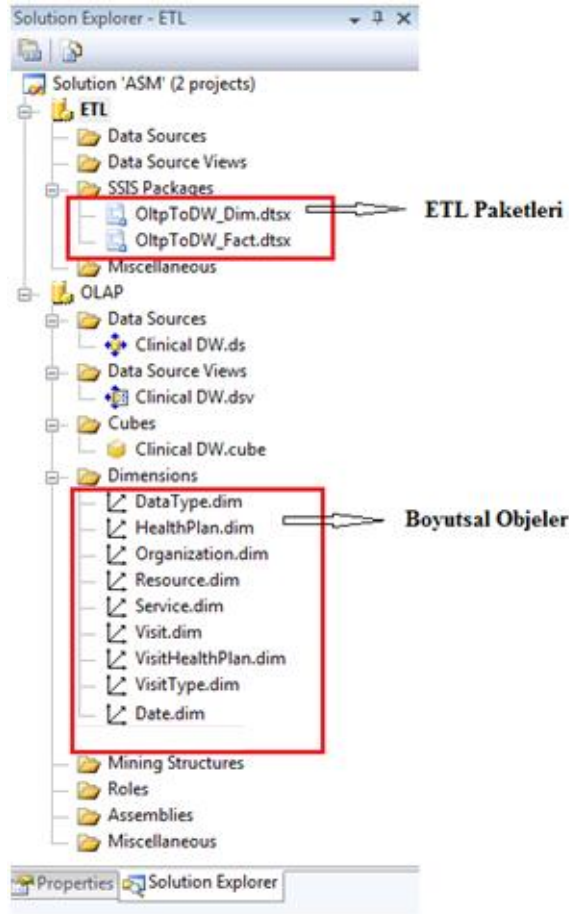
BIDS ortamında bir solution altında birçok farklı iş zekası görevi ya da projesi yer alabilir. Örnek olarak bir solution altında OLTP sistemden verinin çekilip dönüşüm işlemlerinin yapılacağı integration services projeleri ya da veriye çeşitli modeller uygulanarak analiz görevlerinin gerçekleştirileceği analysis services projeleri oluşturulabilir. Son olarak tüm modellerin ve analizlerin kullanılacağı, son kullanıcıya hizmet verecek bir web servisi ya da web sitesi, aynı solution içerisinde barındırılabilir. Programs/Microsoft SQL Server 2008/SQL Server Business Intelligence Development Studio yolu izlenir. BIDS'e ulaşıldığında Şekil 6.2'deki visual studio arayüzü

görüntülenir.



Şekil 6.2 BI uygulama geliştirme ekranı

Solution Explorer Penceresi: Projelerin bulunduğu, yönetim işlemlerinin gerçekleştirildiği penceredir. Oluşturulan BI solution'a yeni projeler buradan eklenebilir ya da isim değişikliği gibi işlemler yapılabilir. Solution Explorer altındaki bir solutionda BI projesinin türüne göre farklı ağaç görünümüne ya da objeler listelenmektedir. Örneğin Şekil 6.3'te olduğu gibi analysis service projeleri boyutsal objeleri içerirken, integration services projeleri ETL paketlerini içermektedir.



Şekil 6.3 BI Solution Explorer

Designer Penceresi: İçindeki objeler veya paketler üzerinde değişikliklerin gerçekleştirildiği alandır. Bu alanda objeler veya paketler üzerinde çalışabilmek için ya yeni bir obje tanımlaması yapılır ya da solution explorer'daki objeleri seçerek değişiklikler gerçekleştirilir.

Pencere Sekmeleri: Pencere sekmeleri, designer pencereleri arasında geçişlerin kolaylıkla yapılmasını sağlar.

Designer Sekmeleri: Bu sekmeler sayesinde; birden çok obje farklı yönlerden düzeltilebilir ya da etkileşime sokulabilir.

Properties Penceresi: Bu pencereden proje ya da solution ile ilgili değişiklikler yapılabilir. Örneğin bir SSAS (SQL Server Analysis Services) objesinin; objectid, dosya adı ve birçok özelliğine bu pencere altından ulaşılabilir ve düzeltmeler yapılabilir.

BI Menüleri: Debug menüsü ile tools menüsü arasındaki menüler BI menüleri olarak isimlendirilmektedir. Örneğin bir SSAS projesindeki Data Source View (DSV) editörünün kullanımında bu alana format ve data source view menüleri eklenecektir.

Output Penceresi: Projenin build ve deploy edilmesi aşamasında mesajların görüntüleneceği kısımdır. Projelerdeki hataların açıklamalı mesajlarına buradan ulaşılabilir.

6.3 Veri Ambarı Tasarımı

Uygulama kapsamında bilgiler şirkete özel olduğu için veriler farklı OLTP sistemlerinden sınırlı bir şekilde excel dosyalarına aktarılarak temin edilmiştir. Excel'e aktarılan bu veriler uygulamanın veri kaynakları olarak kullanıldılar.

Sağlık kurumu için oluşturulacak veri ambarının adı ClinicalDW olarak belirlendi. Kurumun farklı kaynak sistemlerinde bulunan boyut, ziyaret ve işlem verileri veri ambarına yani ClinicalDW ortamına alındı. Tasarım sırasında kaynak sistemlerde bulunan birçok tablo incelenip, analiz edildi. Yapılan analizler sonucunda hangi tabloların kullanılacağına, hangilerinin boyut hangilerinin olgu tablosu olacağına ve tablolarının içeriklerinin ne olacağına karar verildi.

İşlem bilgileri diğer verilerden farklı olarak txt dosyalarında bulunmaktadır. İşlem bilgileri farklı birimlerden farklı formatlarda elde edilmiştir. Bu nedenle farklı txt dosyalarında tutulan işlemler veri ambarında VolumeFact view'ı üzerinden birleştirildi.

6.3.1 Boyut Tabloları

Boyut tabloları oluşturulurken her bir tablonun sonuna Dim yani Dimension (Boyut Tablosu) anlamına gelen kısaltmalar eklenir. Veri ambarı yöneticisi ve kullanıcıların tablolara ilk bakışta ne türde bir tablo olduğunu anlayabilmeleri için bu isimlendirme tekniği kullanılmaktadır.

Tesis

Tablo Adı : FacilityDim

Tesis tablosu sağlık kurumunun farklı lokasyonlarda bulunan şube ve merkezlerinin bilgilerini depolar. Facilityid, tesis tablosunun anahtar alanıdır. Name alanı bu tesislerin

isimlerini gösterir. Numara ile isimlendirilmiş tesislerin isimleri ise shortname alanı ile belirtilir. Tesislerin kendi iç bünyesi dışında bulunan birimlerini göstermek için isexternal alanı kullanılır. Eğer değer “n” ise bu hizmet kurumun kendi hizmetidir, “y” ise dışarıdan alınan bir hizmettir. Kullanılan tesisin aktif olup olmadığı ise active alanının “y” ya da “n” değerlerini almasına göre belirlenir.

	FACILITYID	NAME	SHORTNAME	ISEXTERNAL	ACTIVE
1	1	ASM	3031	n	y
2	2	Suadiye Tıp Merkezi	3032	n	y
3	3	DuzenLab_DZN	DuzenLab_DZN	y	y
4	4	CentroLab_CNT	CentroLab_CNT	y	y
5	5	GENMER_GEN	GENMER_GEN	y	y
6	6	ReferansLab_REF	ReferansLab_REF	y	y
7	7	DuzenIstLab_DIST	DuzenIstLab_DIST	y	y
8	8	Atasehir	3033	n	y

Şekil 6.4 FacilityDim tablosu

Sağlık Güvence Tipi

Tablo Adı : HealthPlanTypeDim

Sağlık güvence tablosu, hastaların sağlık güvencesi tiplerinin tutulduğu tablodur. Sağlık güvence tablosu her güvenceye ait önceden tanımlanmış anahtar alan ve güvence adından oluşur.

	HEALTHPLANTYPEID	NAME
1	1	Özel Sigorta
2	2	Vakıf
3	3	Emekli Sandığı - Devlet Memuru
4	4	Meclis
5	5	SSK

Şekil 6.5 HealthPlanTypeDim tablosu

Anlaşmalı Kurumlar

Tablo Adı : HealthPlanPayerDim

Hastanenin anlaşmalı olduğu sigorta şirketleri, dernekler, vakıflar, odalar, şirketler vardır. Bu tablo sağlık güvence tipi ile ilişkili olarak bu kurumların bilgilerini içermektedir. Description alanında kurumla ilgili tanıtıcı bilgiler yer alır, active alanında ise ilgili kurumların aktif olup olmadığı bilgisi tutulur.

	HEALTHPLANPAYERID	HEALTHPLANTYPEID	NAME	CODE	DESCRIPTION	ACTIVE
1	0	1	PEŞİN	0	NULL	y
2	1	1	Ergo Emeklilik ve Hayat A.Ş.	1	Büyük Mükellefler 4810055642	y
3	2	1	Anadolu Anonim Türk Sigorta Şirketi	2	Büyük Mükellefler 0680061327	y
4	3	1	Demir Hayat Sigorta A.Ş.	3	NULL	y
5	4	1	Allianz Sigorta A.Ş.	4	Büyük Mükellefler 8000013270	y
6	5	1	Yapı Kredi Sigorta A.Ş.	5	Büyük Mükellefler 4560007035	y
7	6	1	Ak Sigorta A.Ş.	6	Bogaziçi Kurumlar 0350003022	y

Şekil 6.6 HealthPlanPayerDim tablosu

Organizasyon Türü

Tablo Adı: OrgTypeDim

Sağlık kurumunun organizasyon bilgilerini içeren tablodur. Kendi içinde hiyerarşik bir yapıya sahip olan bu tabloda orgunittypeid değeri hcu olan kayıttın parentid alanı boştur. Bu organizasyon içerisindeki bütün birimlerin hcu (Hastane) nun altında yer alacağını gösterir.

	ORGUNITTYPEID	NAME	PARENTID	ACTIVE
1	CASHDESK	Cash Desk	HCU	y
2	CLINIC	Clinic	HCU	y
3	DEPT	Department	HCU	y
4	HCU	Hospital	NULL	y
5	IC	Lab Center	IC	y

Şekil 6.7 OrgTypeDim tablosu

Organizasyon Alt Türü

Tablo Adı: OrgSubTypeDim

Bu tablo organizasyon içinde yer alan birimlerin alt kırımlarını tutar. Örneğin, servis tipli bir organizasyonun acil servis ve ameliyat servisi gibi iki alt birimi olduğunu gösterir.

	ORGUNITSUBT...	ORGUNITTYPEID	NAME
▶	ER	SERVCENT	Emergency Room
	OT	SERVCENT	Operating Theater

Şekil 6.8 OrgSubTypeDim tablosu

Kaynak Tipi

Tablo Adı: ResourceTypeDim

Kurumun kaynak tiplerinin tutulduğu tablodur. Bu tabloda her kaynağın bir id'si vardır ve bu id'ye karşılık bir isim bilgisi bulunur. Bu kaynak tipleri hiyerarşik olarak tutulur. Örneğin staff tipinde kendi başına bir kaynak tipi olmasına rağmen, aynı zamanda staff başka kaynak tiplerinin parentid'si de olabilmektedir.

	RESOURCETypeID	NAME	PARENTID	ACTIVE	ISPROVIDER
1	Admitting_Doctors	Admitting Doctors	STAFF	Y	y
2	Anesthetists	Anesthetists	STAFF	Y	y
3	Attending_Doctors	Attending Doctors	STAFF	Y	y
4	BED	Bed	NULL	y	NULL
5	Billing	Billing	STAFF	Y	NULL
6	Blood_Bank_Staff	Blood Bank Staff	STAFF	Y	y
7	Consultants	Consultants	STAFF	Y	y
8	Doctors	Doctors	STAFF	Y	v

Şekil 6.9 ResourceTypeDim tablosu

Kaynaklar

Tablo Adı: ResourceDim

Kaynak bilgilerinin tutulduğu tablodur. Örneğin, doktor Zafer Kamil Berkman staff kaynak tipi altında bulunan bir kaynaktır ve kaynak numarası 2157 dir. Kaynak tiplerinin tutulduğu tabloyla ilişki kurmak üzere tabloda resourcetypeid alanı bulunur.

	RESOURCEID	NAME	DEPTID	LOCATIONID	RESOURCETYPEID
	2157	ZAFER KAMİL BERKMAN	5	1	STAFF
	2158	AYŞE SEDA ÇEVİK	5	1	STAFF
	2159	İSMAİL BURAK UZKAN	3	1	OTHER
	2160	ERHAN TARIM	3	1	OTHER
	2161	ALİYE KOCADAYI	3	1	OTHER
	2162	CENGİZ GÜDÜLOĞLU	5	1	STAFF

Şekil 6.10 ResourceDim tablosu

Servisler

Tablo Adı: ServiceDim

Kurumun bütün tesislerinde verilen servislerin tutulduğu tablodur. Servislerin hangi

departmana bağılı olduğu, servis tipi ve alt tiplerinin ne olduğu bu tablo üzerinde bulunur.

Results

Messages

	SERVICEID	NAME	DEPTID	SERVICSUBTYPEID	SERVICETYPEID	SERVICECENTERID
601	806	Lenfoma Paneli (İmmüfenotiplendir...	102	HEMATOLOGY	LAB	1009
602	807	Akut Lösemi Paneli, kemik iliği	102	HEMATOLOGY	LAB	4583
603	808	Lökosit Alkalen Fosfataz	102	BIOCHEMISTRY	LAB	1009
604	809	Lenfosit (%)	102	BIOCHEMISTRY	LAB	1009
605	810	Plasmodium (Boyalı mikroskopik inc...	102	NULL	LAB	1008
606	811	MCH	102	NULL	LAB	1009
607	812	MCHC	102	NULL	LAB	1009
608	813	MCV	102	NULL	LAB	1009
609	814	Methemoglobin	102	BIOCHEMISTRY	LAB	1009
610	815	Monosit (%)	102	BIOCHEMISTRY	LAB	1009

Şekil 6.11 ServiceDim tablosu

Ziyaret Tipi

Tablo Adı: VisitTypeDim

Sağlık kurumuna gelen kişilerin ziyaret tiplerinin tutulduğu tablodur. Bir hasta ayakta tedavi edilebilir, yatarak tedavi edilebilir ya da acil servise gelebilir vb.

Results		Messages	
	VISITYPEID	NAME	
1	0	Geçici Visit	
2	1	Ayaktan Hasta	
3	2	Yatan Hasta	
4	3	Acil	

Şekil 6.12 VisitTypeDim tablosu

Zaman

Tablo Adı: DateDim

Zaman tablosunun veri ambarı üzerinde oluşturulmasının asıl nedeni performans kaygısını ortadan kaldırmak ve tarih, gün, ay ve yıl parametrelerine daha kolay erişebilmektir. Bu tabloda belli bir tarihin başlangıç değerinden belli bir bitiş değerine kadar tüm günler tutulmaktadır.

Zaman tablosunda, DateKey alanı kayıtlar arasındaki tekliği sağlayan alandır. Zaman tablosu kaynak sistemde bulunmamaktadır. SQL sorguları çalıştırıldığında belli bir

tarihteki Gün (DateKey), Ay (MonthKey), Yıl (YearKey) değerlerine ve o tarihteki aydaki gün sayısı, haftaya ait gün sayısı, günün Türkçe adı “Pazartesi, Salı, .. vb” gibi, ayın Türkçe adı; “Ocak, Şubat, .. vb” gibi, hafta sonu olup olmadığı ve rakamsal değeri diğer tarihlerle karşılaştırma yapabilmek için tutulmaktadır.

	DateKey	Date	DateName	DateNameTR	YearKey	Year	YearName	YearNameTR	HalfYearKey	HalfYear
1	20000101	2000-01-01 00:00:00.000	01-01-2000	01.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
2	20000102	2000-01-02 00:00:00.000	01-02-2000	02.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
3	20000103	2000-01-03 00:00:00.000	01-03-2000	03.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
4	20000104	2000-01-04 00:00:00.000	01-04-2000	04.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
5	20000105	2000-01-05 00:00:00.000	01-05-2000	05.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
6	20000106	2000-01-06 00:00:00.000	01-06-2000	06.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
7	20000107	2000-01-07 00:00:00.000	01-07-2000	07.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
8	20000108	2000-01-08 00:00:00.000	01-08-2000	08.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
9	20000109	2000-01-09 00:00:00.000	01-09-2000	09.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
10	20000110	2000-01-10 00:00:00.000	01-10-2000	10.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00
11	20000111	2000-01-11 00:00:00.000	01-11-2000	11.01.2000	2000	2000-01-01 00:00:00.000	2000	2000	20000101	2000-01-01 00

Şekil 6.13 DateDim tablosu

Örneğin DateKey alanında tutulan 20100805 değeri için ay değeri 08 dir. Operasyonel koşullarda bulunduğunuz aya erişebilmek için SQL sunucusu üzerinde mevcut month(Date) fonksiyonunu kullanmamız gerekir. Fakat SQL sorgularında fonksiyon kullanmak performans açısından olumsuzluk yaratır. Bunun yerine bulunulan tarih DateDim tablosundan çekilir ve month fonksiyonunu kullanmak yerine DateDim tablosundaki MonthKey değeri kullanılabilir.

6.3.2 Olgu Tabloları

Uygulamadaki olgu tabloları ziyaret ve işlem tablolarından oluşmaktadır. Olgu tabloları oluşturulurken her tablonun sonuna fact yani (Olgu Tablosu) anlamına gelen ifadeler eklenir.

Ziyaret Tablosu

Tablo Adı : VisitFact

Ziyaret tablosu, günlük olarak hastaneye gelen hastaların bilgilerinin tutulduğu tablodur. Bu tablo da hastanın ziyaret tipi, hangi hizmet birimine geldiği ve hangi zaman aralıklarında hastanede bulunduğu bilgisi yer alır.

	IVID	IPID	VISITTYPEID	ORGUNITID	STARTDATEKEY	ENDDATEKEY
▶	474501	120192	1	127	20061119	20061119
	474505	120192	1	175	20070820	20070820
	474506	120192	1	127	20071105	20071105
	474524	120193	1	111	20070816	20070816
	474528	120193	1	160	20070917	20070917
	474532	120194	1	127	20050329	20050329
	474556	120194	1	116	20070416	20070416

Şekil 6.14 VisitFact tablosu

İşlem Tablosu

Tablo Adı : VolumeFact

İşlem tablosu hastanede yapılan işlemlerin günlük olarak tutulduğu tablodur.

	SERVICEID	RESOURCEID	IPID	IVID	ORDERID	VISITTYPEID	DATEKEY	SERVICECENTERID	FACILITYID	DataTypeId
1	12221	3030	49817	914038	5854511	1	20090609	1006	1	1
2	12221	3030	121523	593539	2844191	2	20080324	1006	1	1
3	12221	3030	127755	599682	2897474	2	20080331	1006	1	1
4	12221	2741	129711	596456	2867675	2	20080327	1006	1	1
5	12221	3030	115998	650564	3412476	2	20080604	1006	1	1
6	12221	3030	115998	650564	3412568	2	20080604	1006	1	1
7	12222	3788	142859	761832	4443495	2	20081117	1006	1	1
8	12222	3706	127552	792439	4707916	2	20090102	1006	1	1
9	12221	3030	115998	650564	3391715	2	20080604	1006	1	1

Şekil 6.15 VolumeFact tablosu

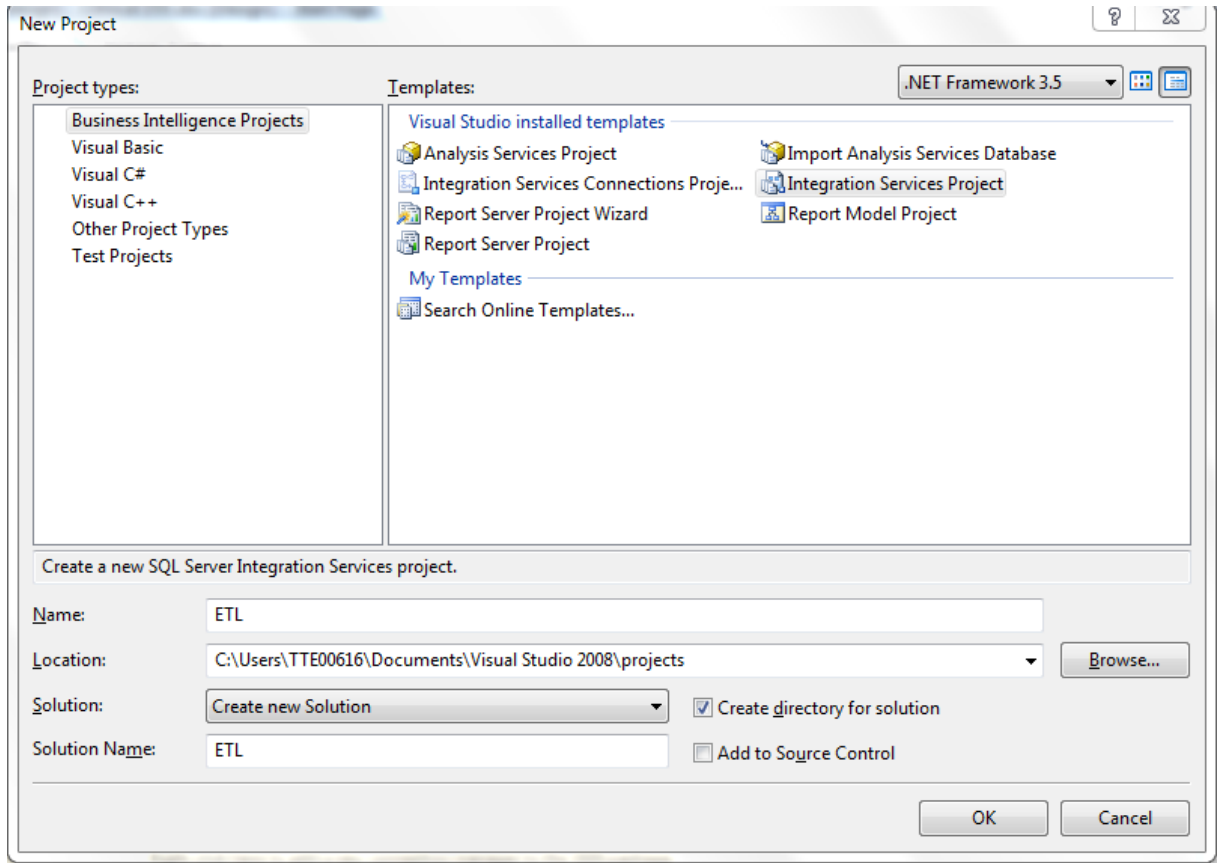
6.4 Veri Aktarım İşlemi

Farklı kaynaklardan elde edilen, değişik şekillerde saklanmış olan veriler, bütünlük içinde işlenebilmek üzere veri ambarına aktarılmalıdır. Yapılan uygulamada veriler SQL Server 2008 SSIS (SQL Server Integration Services) kullanılarak veri ambarına aktarılmıştır. SSIS veri ortamına dışarıdan veri alımını ve farklı ortamlara farklı yapıda veri iletimini sağlayan bir yapıdır. Aktarım aşamasında veri özetleme, dönüştürme ve yükleme işlemleri gerçekleştirilmiştir. ETL işleminde ilk olarak boş kayıtlar yok edilerek veri özetleme işlemi, sonrasında farklı formattaki verileri ortak bir formata getirerek dönüştürme işlemi, son olarak ise hazır verilerin veri ambarına iletilmesi için yükleme işlemi gerçekleştirilmiştir.

Ayrıca yazılım kontrol eksikliklerinden ve kullanıcı giriş hatalarından kaynaklanan veri

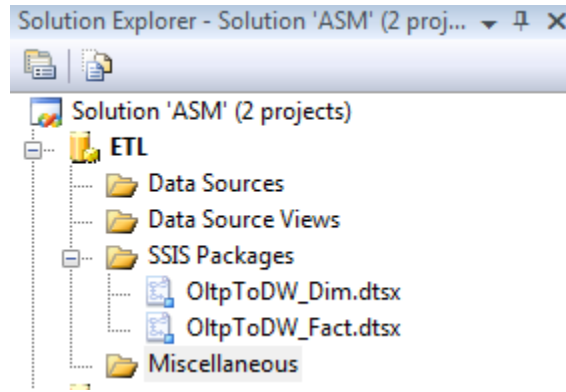
kirliliğini düzeltmek için veri temizleme işlemi yapılmıştır. Örneğin kayıtlar arasındaki eksik ilişkilerin kurulması, eksik veya yanlış alanlara girilmiş adres, telefon vb. bilgilerin formatlarının tutarlı hale getirilmesi.

Daha önce oluşturulmuş olan veri ambarı tablolarını (tabloların yaratılması için kullanılan create scriptleri ek1’de bulunmaktadır.) düzgün, tutarlı, arındırılmış veriler ile yüklemek için bir SSIS projesi yaratmak gerekir. Visual Studio 2008 programında yeni proje oluşturma penceresinde sol paneldeki Business Intelligence Projects seçeneği seçilir. Şekil 6.16’da görüldüğü gibi integration services project seçilip ETL ismi verilerek proje başlatılır.



Şekil 6.16 Integration Services Project oluşturma

Proje ilk açıldığında SSIS Packages altında Package.dtsx isimli dosya otomatik olarak oluşur. Bu dosyanın ismi proje doğrultusunda değiştirilebilir. Uygulama için paketin ismi OltpToDW_Dim.dtsx olarak kaydedilmiştir. Bu paket boyut tablolarını yüklemek için kullanılır. Olgu tablolarını yüklemek için OltpToDW_Fact.dtsx isimli yeni bir paket daha yaratılır (Şekil 6.17) .

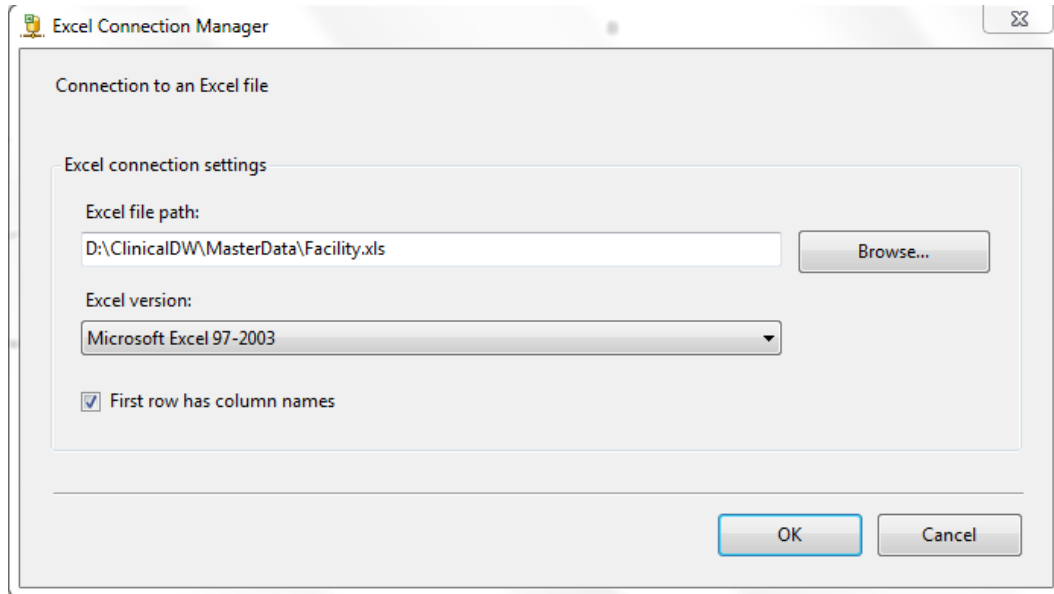


Şekil 6.17 Integration Service Solution Explorer penceresi

Öncelikli olarak farklı paketler tarafından erişilebilecek ortak bir bağlantı oluşturulur. Hedef veri tabanı için ClinicalDW adında bir bağlantı oluşturulmuştur. Böylece kaynak sistem tablolarından datalar alınıp hedef veri ambarına atılırken tekrar bağlantı oluşturmak zorunda kalınmaz. Bağlantı oluşturulduktan sonra OltpToDW_Dim.dtsx paketi açılır. Paket üzerinde süreçler oluşturulurken kullanılan nesneler şunlardır;

Control Flow: Pakette çalışacak olan bir veya daha çok görevi içerir. Data Flow: Kaynak, geçiş ve hedefin belirlendiği ve yönetimlerinin yapıldığı ekrandır. Event Handlers: Çalışma zamanında tetiklenecek olan olayların yönetildiği paneldir. Package Explorer: Paketin görünümünün nasıl olacağını takip edildiği paneldir. Connection Managers: Bağlantı yönetiminin yapıldığı paneldir. Toolbox: Control Flow ve Data Flow için kullanılacak nesneler bütünüdür.

Uygulama için kullanılacak olan kaynaklar Excel ve yazı dosyaları olduğu için bunlara özel bağlantılar kurulmalıdır. Kaynak bilgilerimiz excel dosyalarında bulunur. Kaynak sistem için Connection Managers alanından yeni bir bağlantı yaratılır ve Connection Manager for Excel files bağlantı tipi seçilir. Şekil 6.18'deki ekranda Facility.xls dosyası seçilir ve "First row has column names" seçeneği işaretli bırakılır. Şekil 6.19'da görüldüğü gibi Facility.xls dosyasının ilk satırında sütun isimleri bulunmaktadır.



Şekil 6.18 Excel bağlantı yönetimi

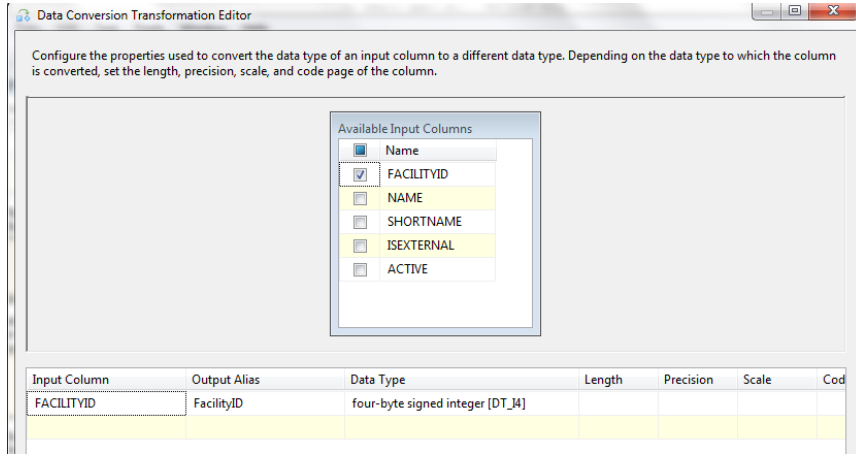
	A	B	C	D	E
1	FACILITYID	NAME	SHORTNAME	ISEXTERNAL	ACTIVE
2	1	ASM	3031	n	y
3	2	Suadiye Tip Merkezi	3032	n	y
4	3	DuzenLab_DZN	DuzenLab_DZN	y	y
5	4	CentroLab_CNT	CentroLab_CNT	y	y

Şekil 6.19 Facility.xls

Bağlantılar oluşturulduktan sonra control flow tabı seçilir, toolboxtan ekrana bir data flow task eklenir, data flow sekmesine geçilir ve excel source kontrolü eklenir. Daha sonra excel source nesnesi seçilerek excel source editor ekranından connection manager nesnesi, o nesneden de Facility bağlantısı ve Sheet1\$ sayfası seçilir.

Veri ambarında bulunan Facility tablosunun facilityid alanı integer tipinde yaratılmıştır, fakat Excel dosyasında bulunan facilityid kolonundaki değerlerden bazıları integer tipinde olmadığı için data aktarımında bu durum sorun yaratacaktır. Bu sorunun giderilmesi için facilityid kolonuna veri dönüştürme işlemi uygulanır.

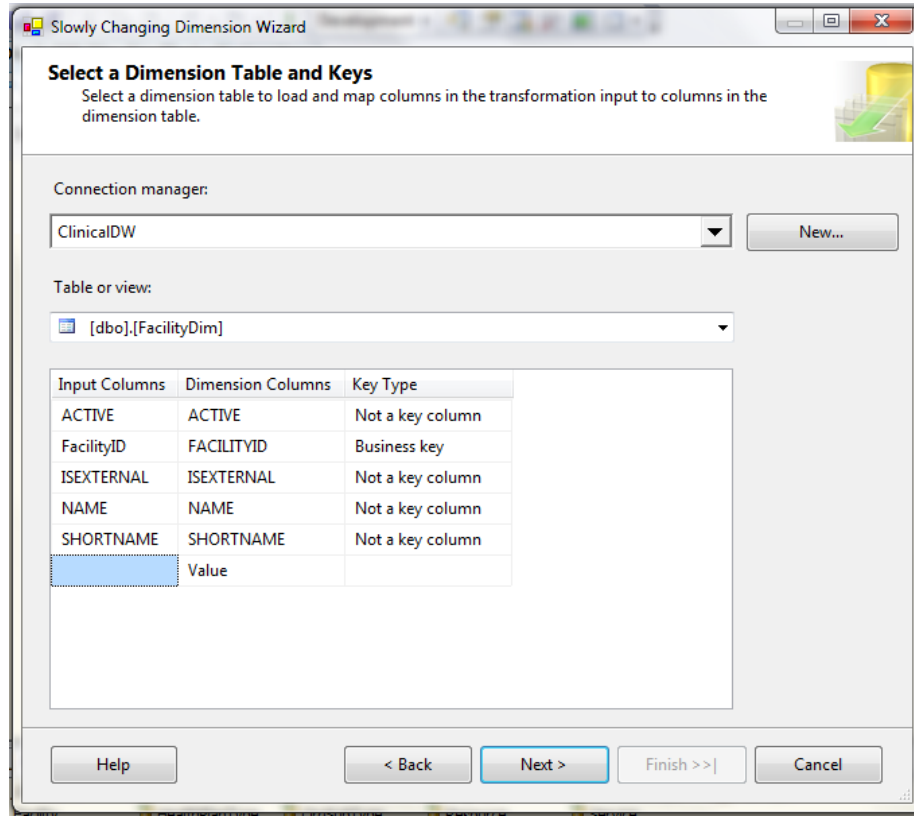
Toolbox'tan pakete bir data conversion nesnesi eklenir. Bu nesne seçilerek Şekil 6.20'deki data conversion transformation editor ekranından facilityid seçilir ve integer tipine çevrilmesi için gerekli kontrol eklenir.



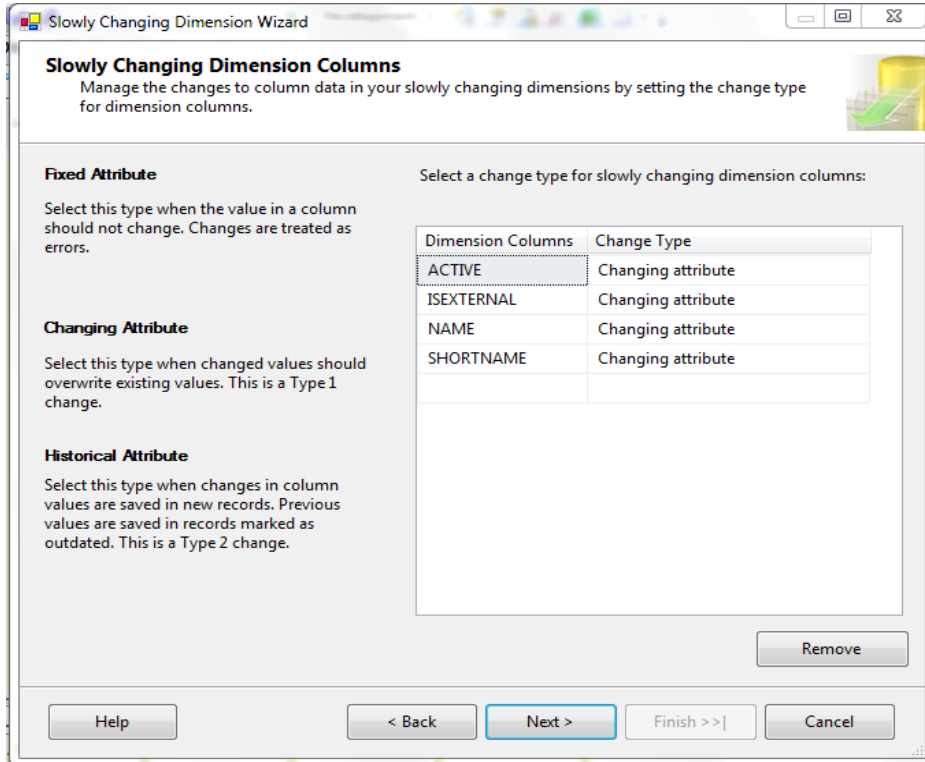
Şekil 6.20 Veri dönüştürme işlemi

Dimension tablolarında bir kaydın güncellenmesi sık rastlanan bir durum değildir, fakat bazı durumlarda ihtiyaç duyulmaktadır. Örneğin kurumların müşteri boyutlarında medeni durumu, yaşadığı şehir gibi özellikler değişebilir. Böyle bir güncellenmenin gerektiği durumlarda boyut tablosuna bu özellik değişimini yansıtmak için slowly changing dimensions tekniği kullanılır. Hastane kurumunun yeni bir tesis açma ya da mevcut bir tesisin ismini değiştirme ihtimali olduğu için boyut tablolarına bu teknik uygulanmıştır.

Toolbox'tan ekrana slowly changing dimension nesnesi eklenir ve bu nesne seçilerek Şekil 6.21'deki slowly changing dimension wizard ekranına gidilir. Bu ekranda hedef tablo FacilityDim tablosunu seçmek için ClinicalDW bağlantısı seçilir. Facilityid iş anahtarı olarak seçilir ve bir sonraki ekrana geçilir. Şekil 6.22'deki ekranda active, isexternal, name, shortname alanlarından biri değiştiğinde, tabloda ilgili facilityid varsa güncelleme yapılacağı, yoksa yeni bir kayıt ekleneceği belirtilir.

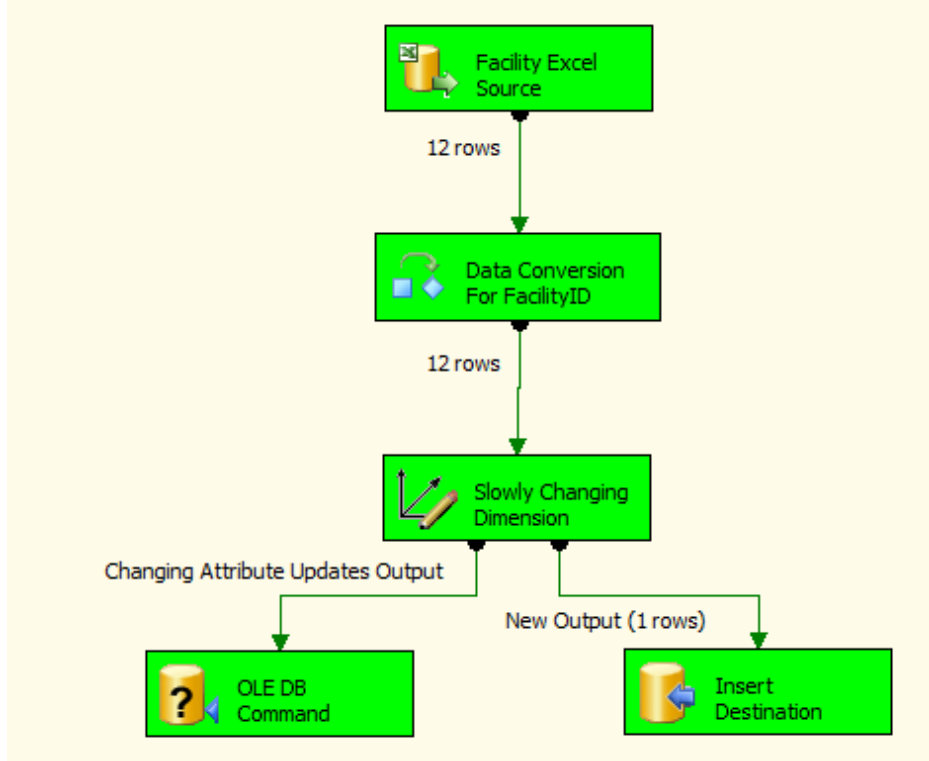


Şekil 6.21 Slowly changing dimension wizard



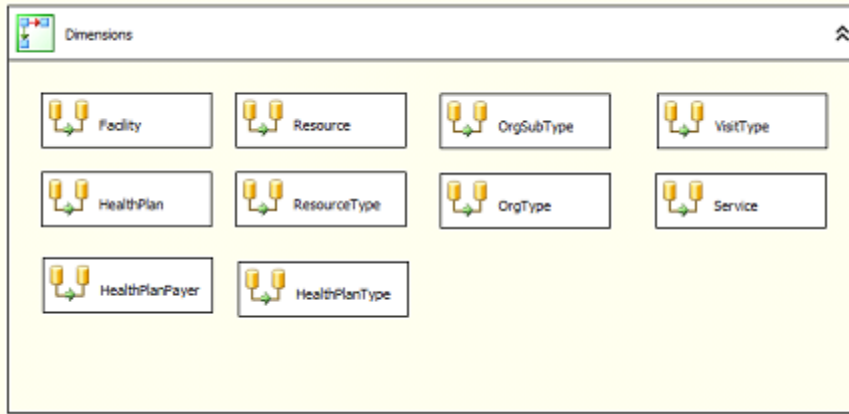
Şekil 6.22 Slowly changing dimension boyut seçimi

Bütün bu işlemlerden sonra excel dosyasından verileri alınıp, dönüştürüp sonra da hedef tabloya kayıtları atacak bir flow tasarlanmış duruma gelinir. Tasarlanan bu akış çalıştırıldığında Şekil 6.23'deki gibi bütün adımlar yeşil renk olur. Özetle tasarlanan akışın başarılı bir şekilde tamamlandığı anlaşılır.



Şekil 6.23 ETL paketinin akış şeması

Facility için uygulanan işlemler sırası ile resource, OrgSubType, VisitType, HealtPlan, ResourceType, OrgType, Service, HealtPlanPayer, HealtPlanType boyutları için uygulanır. Daha sonra bu data flow nesnelerini bir arada çalıştırabilmek için Şekil 6.24'deki sequence container kullanılır. Böylece OltpToDW_Dim.dtsx paketi tamamlanır.



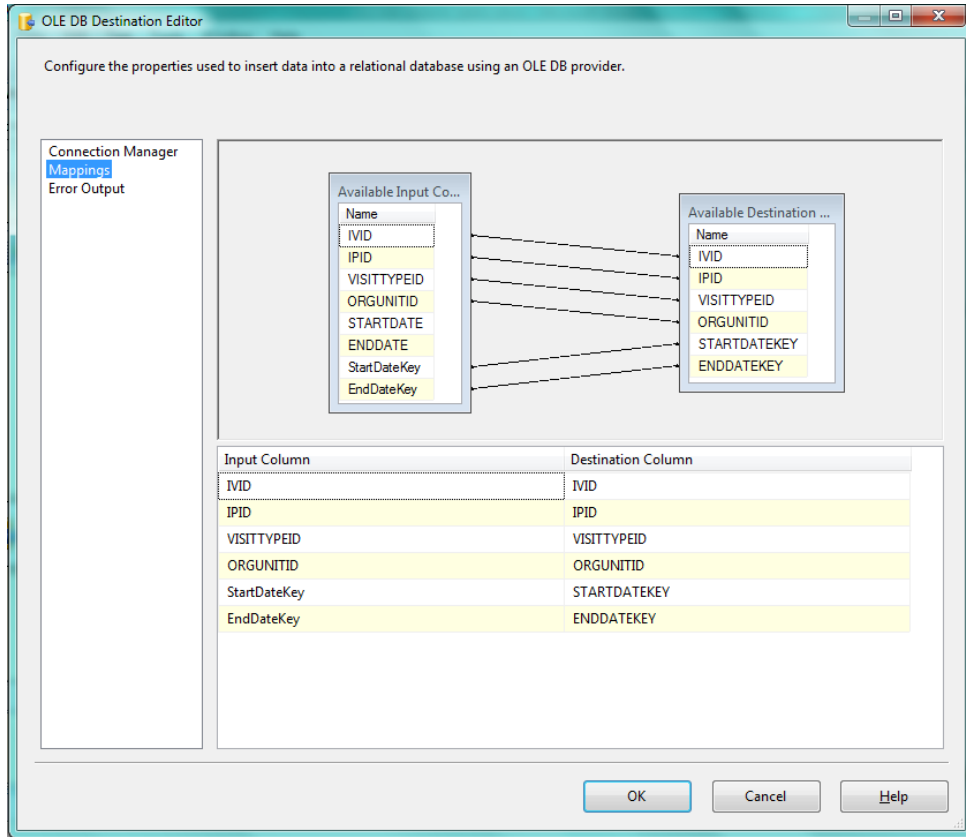
Şekil 6.24 Sequence container

Olgu Tablosu

Boyut tabloları için OltpToDW_Dim.dtsx paketi tamamlandıktan sonra olgu tabloları için yaratılmış olan OltpToDW_Fact.dtsx paketi açılır.

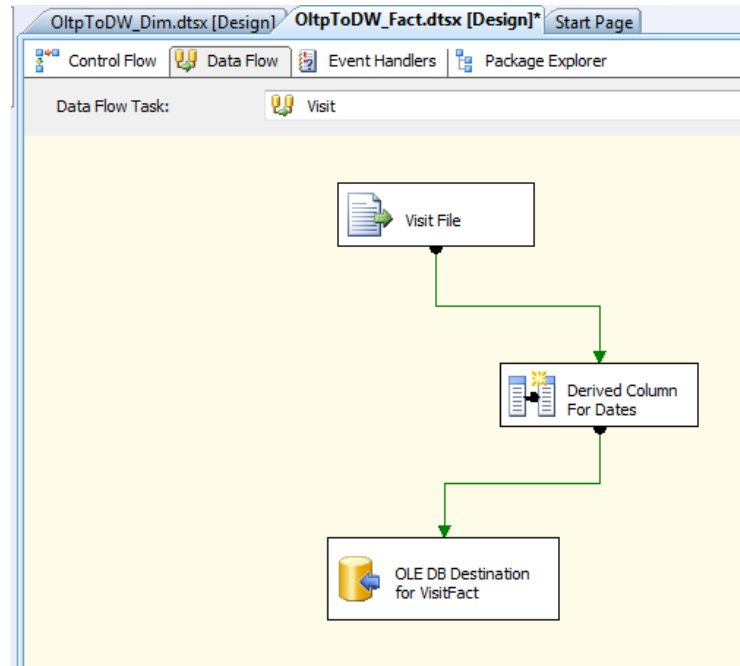
Sağlık kurumu ziyaret verileri her gün güncelleneceği için kayıtların atılacağı tabloda da verinin güncellenmesi gerekmektedir. Öncelikle akışa bunun için bir kontrol eklenmelidir. Boyut tablolarını dolduran pakette yapıldığı gibi hedef tablolar için connection managers üzerinden ClinicalDW bağlantısı yapılır. Control flow ekranında iken, ekrana Execute SQL Task kontrolü eklenir. Oluşan kontrolün ismi Delete Visit Table olarak değiştirilir. Daha sonra control flow ekranında oluşturulan Delete Visit Table seçilerek Execute SQL Task editörü açılır. Şekil 6.25'te görüldüğü gibi bu SQL sorgusunu çalıştırmak için bağlantı ClinicalDW olarak seçilir. Sonra SQL Statement kısmına çalıştırılmak istenen SQL cümleciği yazılır. Böylece bu akış her çalıştığında ilk olarak Visit tablosundaki verileri temizleyecek hale gelmiştir.

Araç kutusundan OLE DB Destination kontrolü data flow ekranına eklenir. OLE DB destination for VisitFact olarak isimlendirilen bu kontrol seçilerek OLE DB connection manager listesinden daha önce yaratılmış olan ClinicalDW bağlantısı ve verinin aktarılacağı Visit tablosu seçilir. Son olarak mapping sayfasına gidilerek, hangi alanın tabloda hangi alana karşılık geleceği eşleştirmesi yapılır. Kolon isimleri aynı ise, Şekil 6.27'deki gibi integration services otomatik olarak kolon eşlemelerini gerçekleştirecektir.



Şekil 6.27 Alan eşleştirmelerinin yapılışı

Bütün bu işlemlerden sonra yazı dosyasından veriler alınıp, dönüştürülüp sonra da hedef tabloya kayıtları atacak bir akış tasarlanmış olur. Tasarlanmış bu akışı çalıştırdığımızda bütün kontroller yeşil renk olur. Böylelikle hedef yapılandırması tamamlanmış olur.



Şekil 6.28 Olgu tablosu için paket akışı

Visit için uygulanan işlemler aynı şekilde Volume tablosu için de uygulanır. Daha sonra bu kontroller Fact Tables isimli sequence container içine alınarak OltptoDW_Fact.dtsx paketi tamamlanır.

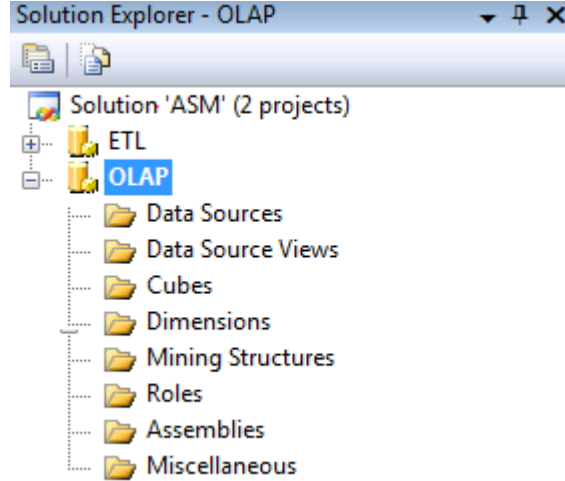
6.5 Küp Tasarımı

Sunucuda ClinicalDW veri ambarı için boyut ve olgu tabloları yaratılıp, bu tabloların ETL işlemleri ile doldurulmasından sonra OLAP küplerini içerecek analiz veritabanının oluşturulması işlemine geçilir. Çalışmada analiz sistemi olarak Microsoft Analysis Services kullanılmıştır. Microsoft SQL Server 2008 R2 Analysis Services, OLAP ve data mining için bir ara katmandır. Hızlı kullanıcı erişimi ve veri analizi için çok boyutlu küpleri oluşturan ve yürüten analysis server'ı içerir. Analysis services kompleks analitik sorgulara hızlı cevap verebilmek için veri ambarındaki verileri, önceden hesaplanmış veriler olarak küpler içerisinde organize eder.

6.5.1 Projenin Oluşturulması

Analysis services projesi SQL Server Business Intelligence Development Studio ortamında oluşturulur. Proje tiplerinden Business Intelligence Projects - Analysis Services Project seçilerek ASM projesi yaratılır.

Şekil 6.29’da görüldüğü gibi solution explorer da ASM projesine Analysis Services OLAP projesi eklendiğinde Data Sources, Data Source Views, Cubes, Dimension, Mining Structures, Roles, Assemblies ve Miscellaneous klasörleri otomatik olarak eklenir.



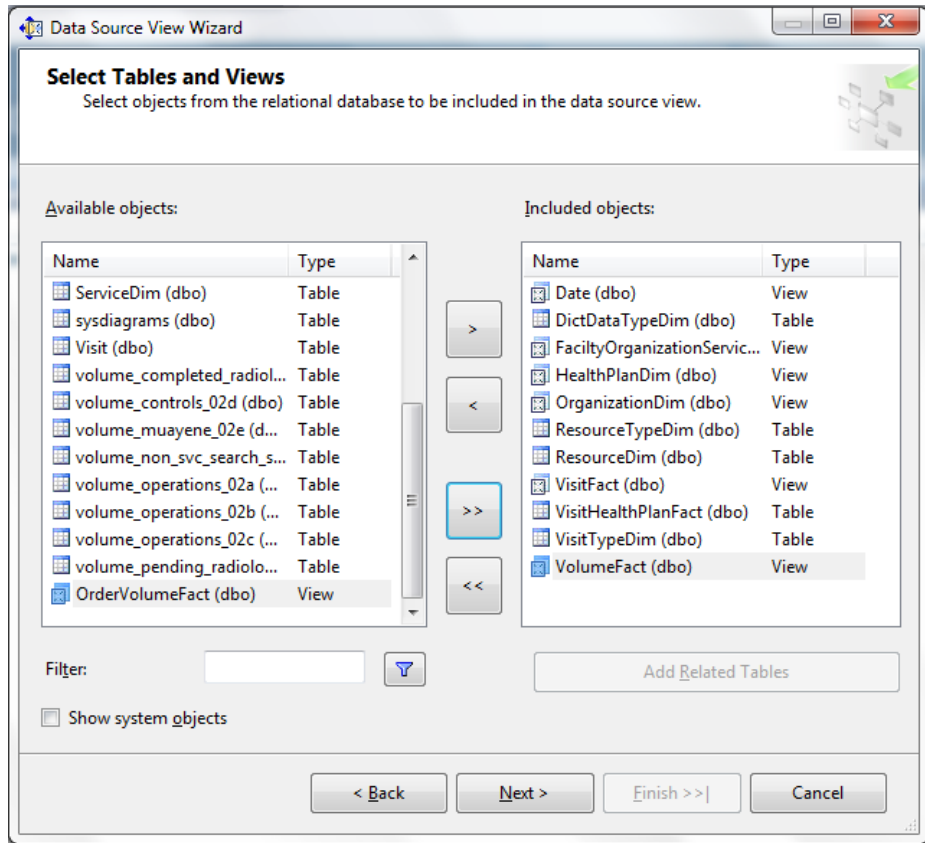
Şekil 6.29 OLAP projesinin nesneleri

6.5.2 Data Source Tanımlanması

Bu bölümde uygulamada kullanılacak olan veri tabanı seçilir. Bunun için Solution Explorer’da data sources nesnesi üzerinden gidilerek açılan sihirbazda create new connection seçeneği seçilir. Açılan connection manager penceresinde bağlantı için provider olarak “Native OLE DB\SQL Server Native Client 10.0”, sunucu olarak yerel bilgisayar kullanıldığı için “.R2” ve veritabanı olarak da daha önce yaratılmış olan “ClinicalDW” veritabanı seçilir. Kurulan bu yeni bağlantının doğruluğunu test etmek için test connection butonu kullanılır.

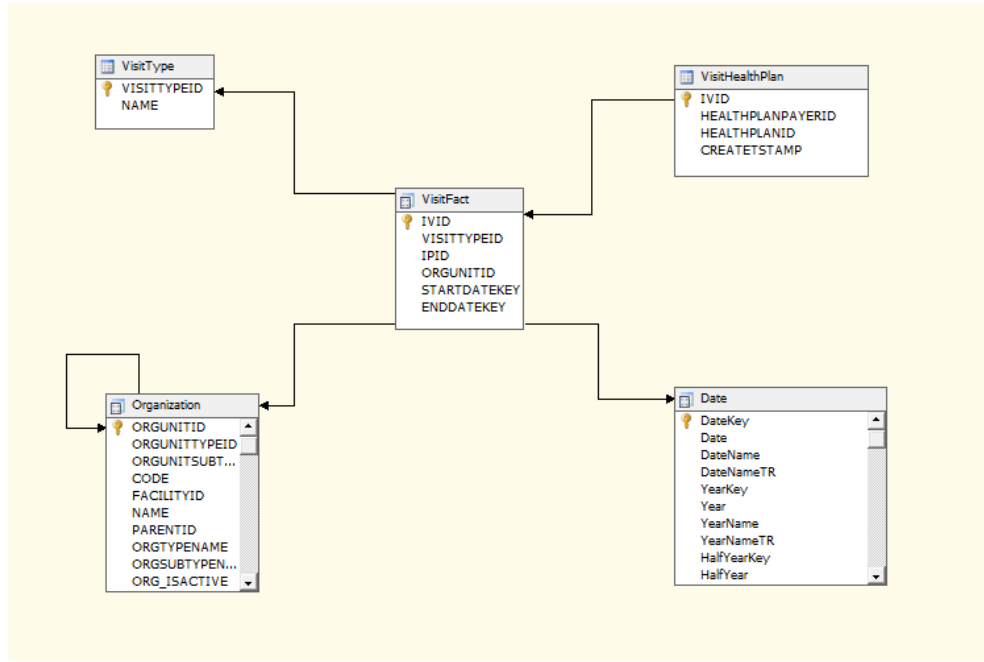
6.5.3 Data Source View Tanımlanması

Data Sources View objesi için yeni bir bağlantı yaratılır. Bu bağlantı için sunucu olarak yerel bilgisayar kullanıldığı için “.R2” seçilir ve veritabanı olarak da daha önce yaratılmış olan “ClinicalDW” veritabanı seçilir. Daha sonra şekil 6.30’da görüldüğü gibi küp için gerekli olan olgu ve boyut tabloları, veri ambarından seçilerek data source view işlemi tamamlanır.



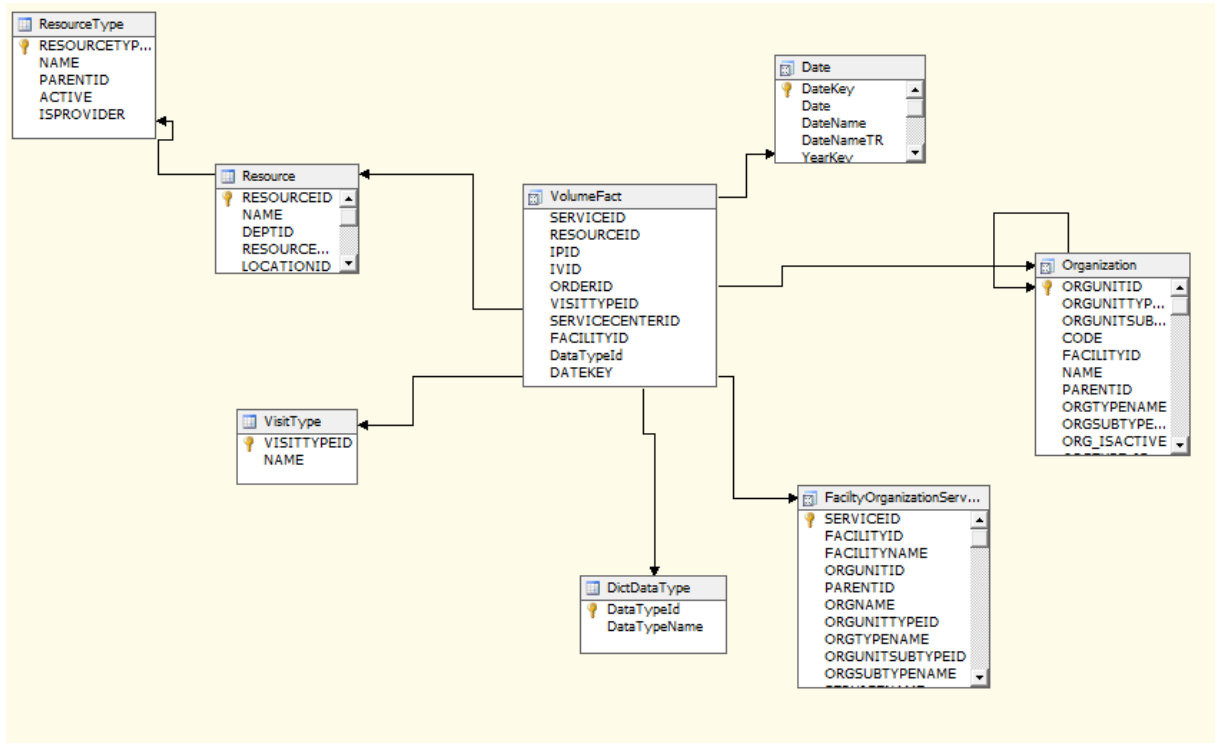
Şekil 6.30 Küp için gerekli tabloların seçimi

Ziyaret değerleri için yıldız şema modeli tercih edildi. Şekil 6.31’de görüldüğü gibi oluşturulan visit yıldız şemasının merkezinde VisitFact olgu tablosu yer alır. Bu tablo, çok boyutlu analizlerde incelenecek olan gerçek ziyaret verilerini depolar. Visitfact tablosunun içinde boyut tabloları ile ilişkiyi sağlayacak ivid, visittypeid, orgunitid, startdatekey, enddatekey anahtar alanlar bulunur. Bu anahtarlarla boyut tabloları merkez olgu tablosuna bağlanır. Bu yapı VisitFact olgu tablosundaki ölçüm değerlerini ziyaret tipine, sağlık güvence tipine, hastane bölümüne ve zamana göre inceleme imkanı yaratır.



Şekil 6.31 Ziyaret bilgileri için yıldız şema modeli

İşlem bilgileri için kar tanesi modeli tercih edildi. Şekil 6.32’de görüldüğü gibi oluşturulan işlem bilgileri kar tanesi şemasının merkezinde VolumeFact olgu tablosu yer alır. Bu tablo, çok boyutlu analizlerde incelenecek olan hastanede yapılan gerçek işlem verilerini depolar. VolumeFact tablosunun içinde boyut tabloları ile ilişkiyi sağlayacak serviceid, resourceid, visittypeid, datatypeid, orgunitid, datekey anahtar alanları bulunur. Bu model VolumeFact olgu tablosundaki ölçüm değerlerini ziyaret tipi, hastane birimi, zaman ve kaynak tipi kırılımında inceleme imkanı yaratır. Eğer bu şema modeli tercih edilmeseydi; doktor, anestezi uzmanı, danışman, laborant gibi kaynak tipi bazında analizler yapılamazdı.

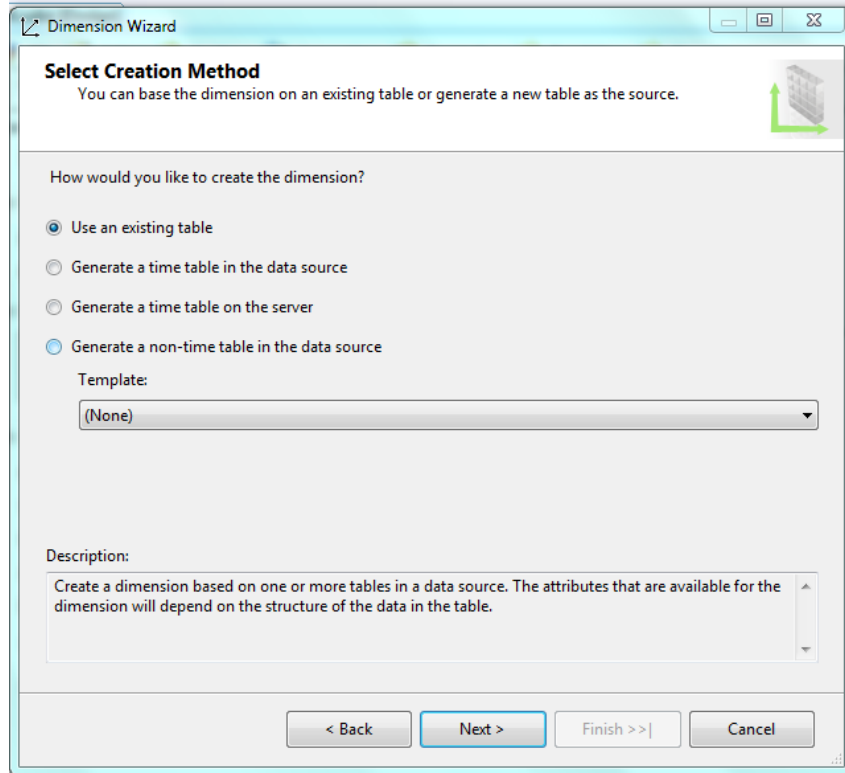


Şekil 6.32 İşlemler için kar tanesi şema modeli

6.5.4 Küp Boyutlarının Oluşturulması

Küp oluşturulmadan önce küpte yer alacak boyutlar oluşturulur. Yapılan uygulamada ziyaretlerin ve işlemlerin hangi boyutlara göre inceleneceği analiz edilerek küp boyutları olarak Date, DataType, HealthPlan, Organization, Resource, Service, VisitHealthPlan, VisitType yaratıldı.

Kullanıcılar veriyi zamana bağlı olarak özel bir hiyerarşi içerisinde görmek istemektedirler. Verinin zamana bağlı analizinin gerçekleştirilmesi için bir küpte zaman boyutunun oluşturulması gerekmektedir. Zaman boyutu boyut sihirbazı kullanılarak oluşturulmaktadır. Boyut sihirbazı açıldığında kullanıcının karşısına Şekil 6.33'te görülen iletişim kutusu gelmektedir. Bu seçim yapıldıktan sonra Şekil 6.34'de olduğu gibi boyut tablosu seçimi iletişim kutusundan zaman bilgisinin yer aldığı tablo seçilerek zamanla ilgili veriler alınmaktadır.



Dimension Wizard

Select Creation Method
You can base the dimension on an existing table or generate a new table as the source.

How would you like to create the dimension?

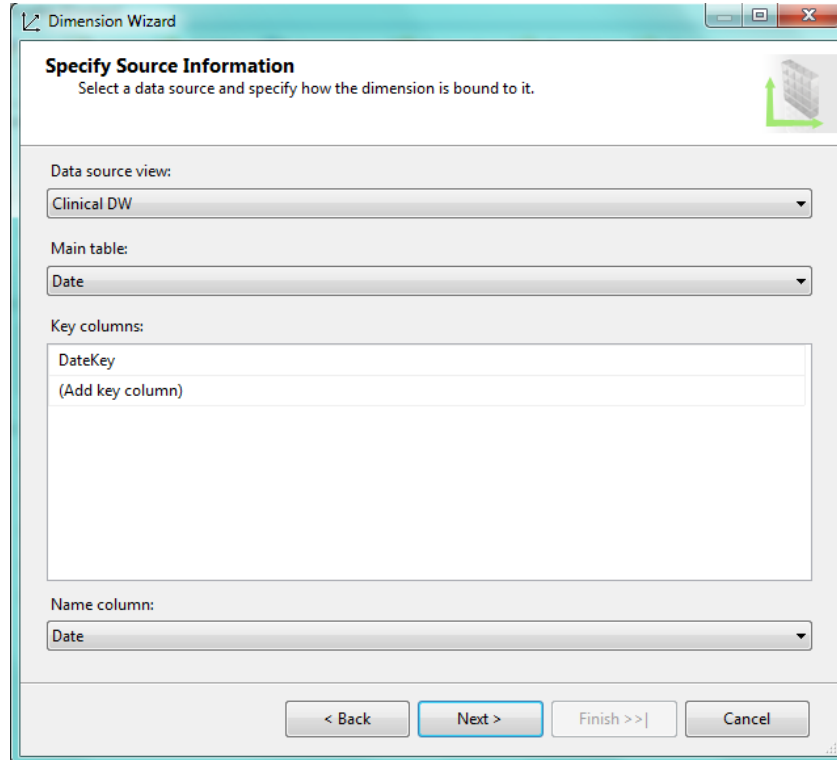
☒ Use an existing table
☐ Generate a time table in the data source
☐ Generate a time table on the server
☐ Generate a non-time table in the data source

Template:
(None)

Description:
Create a dimension based on one or more tables in a data source. The attributes that are available for the dimension will depend on the structure of the data in the table.

< Back Next > Finish >> | Cancel

Şekil 6.33 Boyut yaratma sihirbazı



Dimension Wizard

Specify Source Information
Select a data source and specify how the dimension is bound to it.

Data source view:
Clinical DW

Main table:
Date

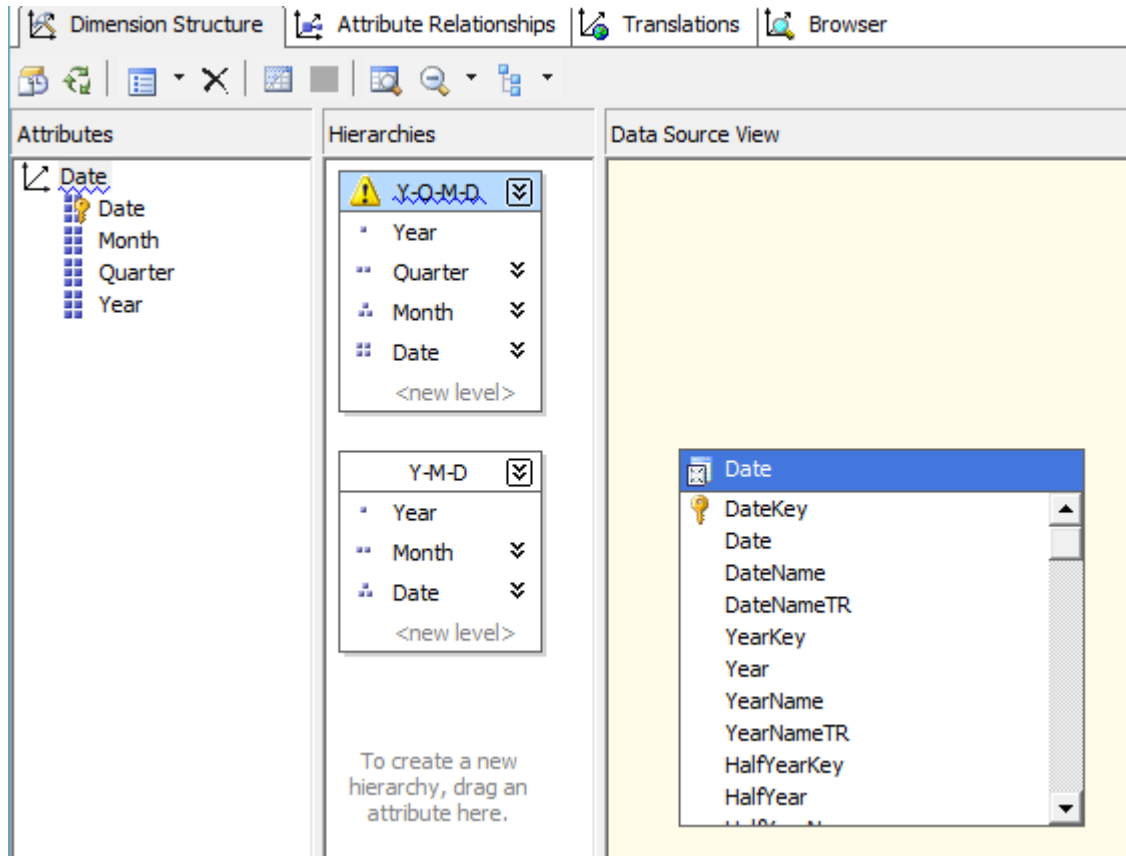
Key columns:
DateKey
(Add key column)

Name column:
Date

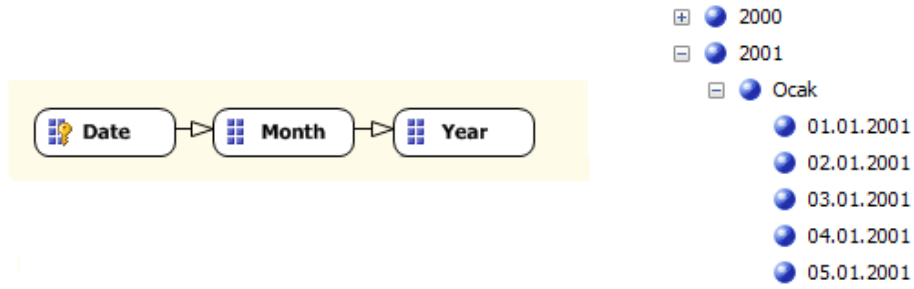
< Back Next > Finish >> | Cancel

Şekil 6.34 Date boyutu için tablo ve key alan seçimi

Date boyutu yaratıldıktan sonra bu boyuta yeni bir özellik eklenmek istendiğinde, istenilen özellik dimension structure sekmesinde bulunan data source view alanından attributes alanına sürüklenir. Şekil 6.35'te görüldüğü gibi analizler gün, ay, yıl ve çeyrek bazında yapılacağı için Date boyutuna day, month, quarter, year özellikleri eklendi. Daha sonra bu özelliklerden iki hiyerarşi oluşturuldu. Aslında hiyerarşiler, birbiri ile ilişkili değerlerin tutulduğu büyük bir boyut olarak düşünülebilir. Şekil 6.36'da Y-M-D hiyerarşisinin özelliklerinin birbiriyle ilişkisi attribute relationships sekmesinde tanımlanır ve browser sekmesinden özelliklerin verileri görülür.



Şekil 6.35 Tarih boyutunun detayları

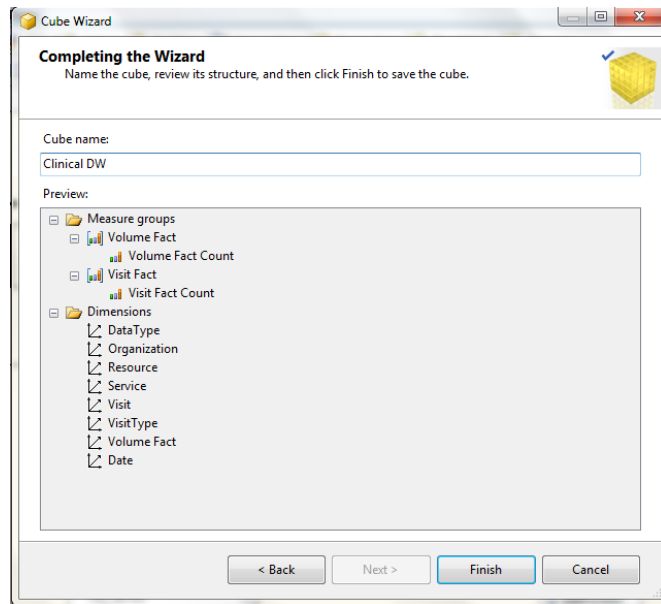


Şekil 6.36 Hiyerarşi ilişkileri ve boyut verilerinin görünümü

Aynı şekilde diğer boyutlar için istenilen özellikler eklenip hiyerarşiler oluşturularak boyut oluşturma işlemi tamamlanır.

6.5.5 Küpün Oluşturulması

Yeni bir küp oluşturmak için öncelikle veri kaynağı seçilir. Yaratılmış olan ClinicalDW bağlantısı seçilerek ilerlenir. Küp sihirbazı yardımıyla tablolar kısmından olgu kısmına VisitFact ve VolumeFact tabloları seçilir. Aynı şekilde boyut bölümü için ise HealthPlan, Organization, Resource, Service, VisitHealthPlan, VisitType tabloları seçilir. Analiz konusunun niceliğini belirleyen ölçümler ise Volume Fact Count ve Visit Fact Count olarak belirlenir. Oluşturulan küpün yapısı küp editöründe Şekil 6.37’de olduğu gibi görüntülenir.



Şekil 6.37 Küp için gerekli olgu ve boyut değerlerinin belirlenişi

Dimension usage sekmesinden daha önceden oluşturulmuş olan ölçümler ve boyutlar arasındaki ilişkilerin türleri belirlenerek (regular, fact, referenced, many-to-many, datamining, no relationship) çok boyutlu ilişkiler kurulur. Organization boyutu ile hem işlem hem de ziyaret ölçümleri regular olarak ilişkilendirilir. Diğer ilişkilerden farklı olarak Healtplan ile Volume ölçümü arasındaki ilişki ele alınabilir. Bu ilişki many-to-many ilişkisidir. Bir tane sağlık güvencesi olan hasta birden fazla işlem yapabilir ya da birden fazla sağlık güvencesi olan hasta tek bir işlem yapabilir. Bu ilişkiyi operasyonel veritabanının da yapma şansı yoktur. Bize bu esnekliği sunan küp yapısıdır.

Measure Groups		
Dimensions	Volume	Visit
Organization	OrganizationId	OrganizationId
HealthPlan	Visit Health Plan	Visit Health Plan
Visit	IVID	IVID
Resource	ResourceId	
Date	Date	Date

Şekil 6.38 Boyut ile ölçümler arasındaki fiziksel ilişkiler

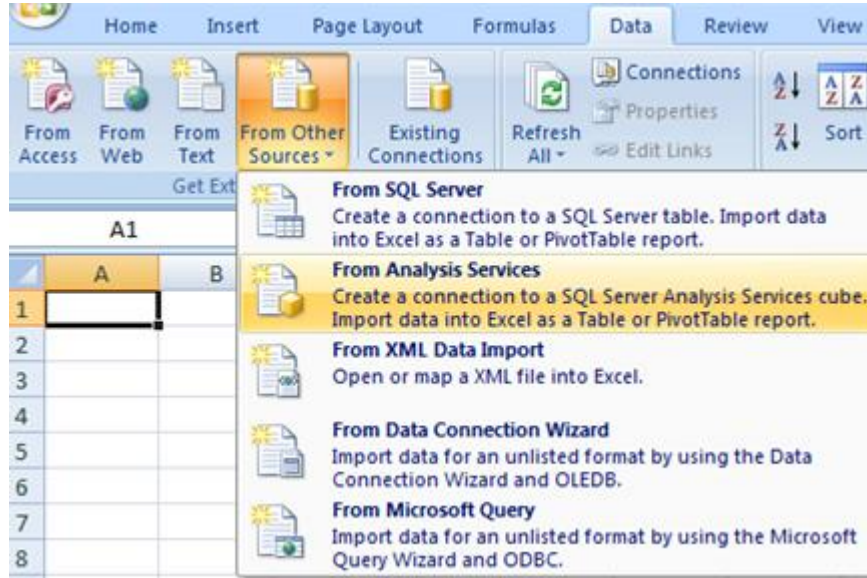
Calculation sekmesinde hesaplamalar oluşturulur. Bu uygulama için visit adında $[Measures].[Visit\ Count]/[Measures].[Volume\ Count]$ ifadesiyle bir hesaplama yapılmıştır. Son olarak Şekil 6.39'daki gibi browser sekmesinden küpteki verilere detaylı kırılımlarla bakılabilir.

6.6 MS Excel 2007 ile OLAP Küp Raporlaması

Kullanım yaygınlığı ve lisans sorununun olmaması nedeniyle, OLAP küplerinin oluşturulmasından sonra kullanıcıların küpü rahatça kullanabilmesi, grafik ve rapor desteği alabilmesi, görünümde değişiklik yapabilmesi için hem istemci hem de sunucu olarak MS Excel 2007 programı kullanılmıştır.

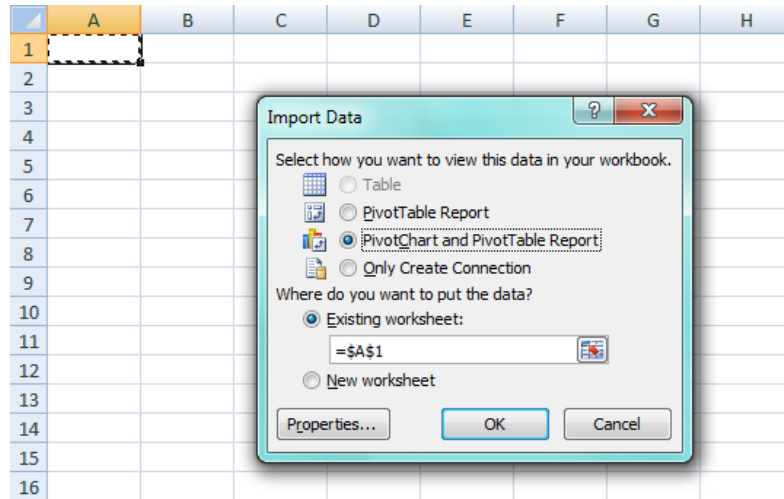
MS Excel, Microsoft şirketi tarafından yazılan ve dağıtımı yapılan bir hesap çizelgesi programıdır. Excel OLAP küplerini oluşturmakta ve kendi içindeki özet tablo ve grafik işlevlerini kullanarak verileri görüntülemekte, veriler üzerinde dolaşabilmeyi sağlamaktadır. Özet tablo işlevi sayesinde veriler büyük ve etkileşimli bir çarpaz tablo şeklinde görüntülenmektedir.

Microsoft Office Excel 2007'de, veri sekmesi altında bulunan dış veri al grubundaki Analysis Services seçilerek bağlantı ekranı açılır. Bu ekran da bağlanılmak istenen sunucunun ismi yazılır ve bir sonraki ekrana geçilir. Sunucuda bulunan ClinicalDW küpü seçilir.



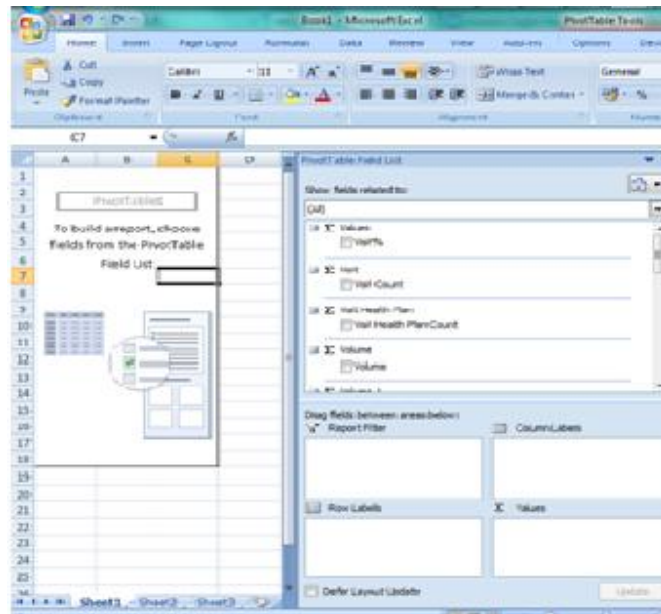
Şekil 6.40 Excel 2007 ile SQL Server 2008 R2 Analysis Services'e bağlantı

Bağlantı kurulduktan sonra Import Data iletişim kutusu açılır. Şekil 6.41'de görüldüğü gibi küp raporlaması yapılacağı için PivotChart and PivotTable Report seçeneği tercih edilir. Ayrıca bu raporun excelin hangi hücrelerinde açılmak istendiği belirtilir.



Şekil 6.41 Özet tablo ve özet grafik sihirbazı

Uygulama sonucunda elde edilen OLAP küpünün Excel'deki özet tablo biçimindeki gösterimi Şekil 6.42'deki gibidir. Elde edilen etkileşimli tablo sayesinde boyutların yerlerini değiştirebilir, detay açma, detay kapama, dilimleme ve kesme işlemleri yapılabilir.

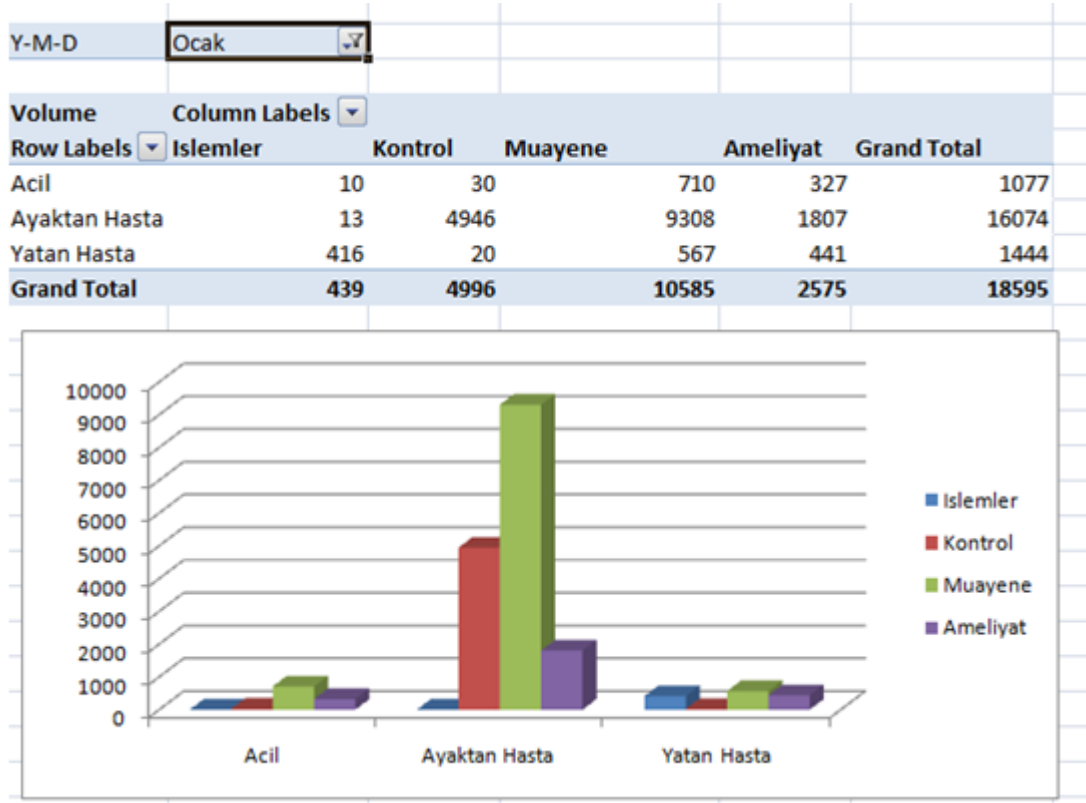


Şekil 6.42 Verilerin özet tablo biçimindeki gösterimi

Bağlantı oluşturulduktan sonra ekranın sağ tarafında PivotTable Field List penceresi belirir. Bu pencere küp üzerindeki bütün boyutları ve ölçümleri gösterir. Hangi ölçüm hangi kısımlarda incelenmek isteniyorsa pencerenin altında bulunan bölümlere sürüklenir.

Excel'e alınan küp datası proje analizinde yöneticilerin, analistlerin ve kullanıcıların veriyi görmek istedikleri şekillerde sunacak olan raporların tasarımında kullanılır. Uygulama kapsamında kurumun öncelik verdiği ilk faz raporları hazırlanmıştır. Aşağıda bu raporlar hakkında detaylı bilgiler bulunmaktadır.

1) Ziyaret tiplerine göre işlem analizleri raporu:



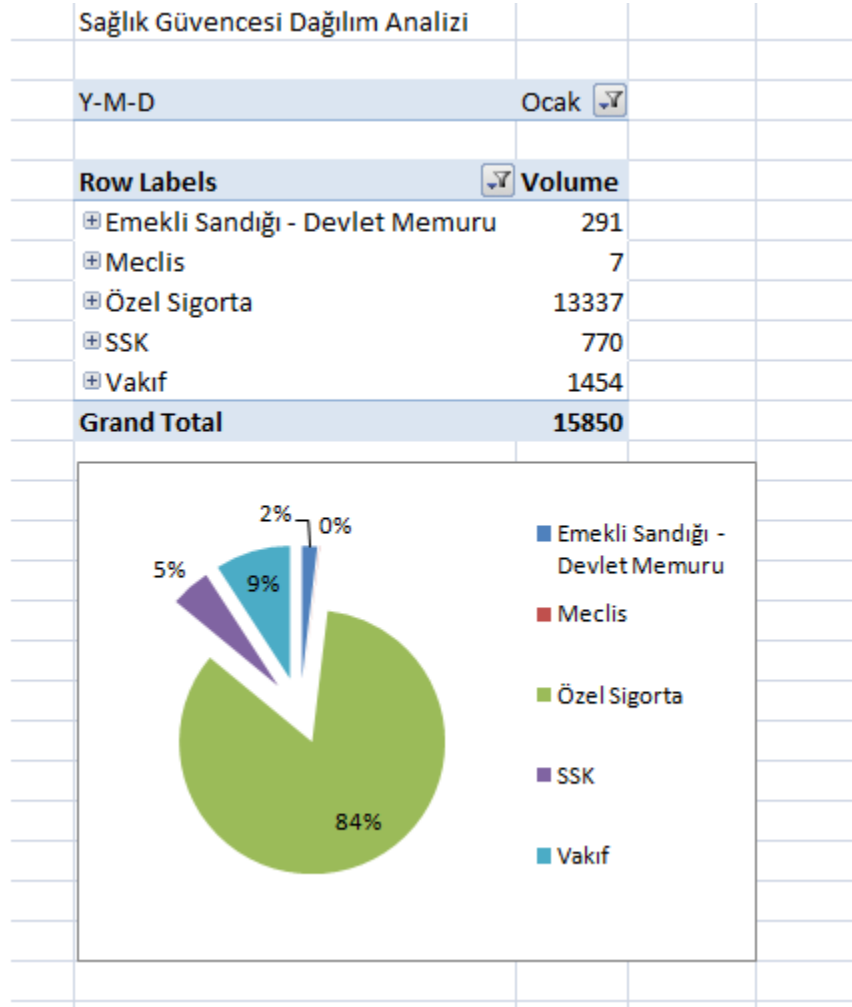
Şekil 6.43 Ziyaret tiplerine göre işlem analizleri raporu

Rapor datayı crosstab ve grafik olarak sunmaktadır. Bu analizler tarih filtresine göre yapılacak şekilde tasarlanmıştır. Kullanıcı tarih boyutuna tanımlanan hiyerarşi yardımı ile karşılaştırmalarını yıllık, aylık ve günlük olarak yapabilir.

Örneğin Ocak ayında Acil ziyaret tipi ile kuruma 1077 hasta gelmiştir. Bu hastaların 30'u kontrol işlemi, 710'u muayene işlemi, 327'si ameliyat işlemi ve 10'u genel işlemi yaptırmışlardır. Rapor incelendiğinde kuruma başvuran hastaların %86'sını ayakta hasta grubu oluşturmaktadır. Hastaneye başvuran acil hastaların ameliyat edilme oranı %30 olarak ölçümlenmiştir. Bu durumda veri ambarından detay bilgilere inilerek acil hastaların kuruma getiriliş zamanları ve oldukları

ameliyat türleri incelenerek bu kritik hastalar için optimum çözümler üretebilecek stratejiler belirlenir.

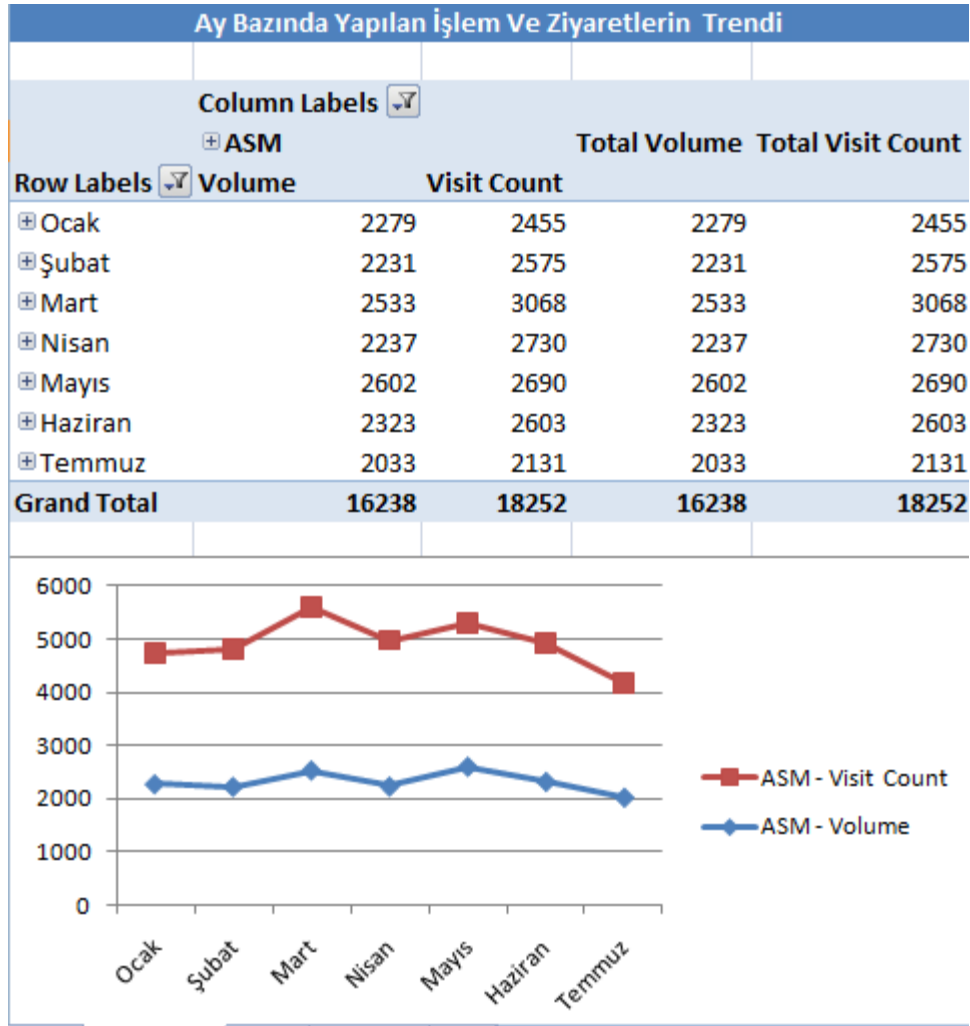
2) Hastaların sağlık güvencesi dağılımı analizi raporu:



Şekil 6.44 Hastaların sağlık güvencesi dağılımı analizi raporu

Bu rapor da tarih filtresine göre çalışmaktadır. Seçilen tarihte kurumda gerçekleşen işlemlerin sağlık güvencesine göre değerlerini listeler. Pie chart grafiğinden anlaşılacağı gibi kurumda işlem yaptıran hastaların %84'ü özel sağlık sigortası olan kişilerdir. HealthPlanType tablosundan beslenen sağlık güvencesi boyutunda datalar özel sağlık kurumlarının detayına kadar tutulmuş olsaydı bu %84'lük grubun sigorta şirketlerindeki dağılımını da ölçümlüyor olurduk.

3) İşlem ve ziyaret bilgilerinin trend analiz raporu:



Şekil 6.45 İşlem ve ziyaret bilgilerinin trend analiz raporu

Genellikle ay bazında yapılan bu trend analizi kurumun aylık performansını ve durumunu görebilmesi ve bu bilgiler ışığında istatistiksel analizler yardımı ile gelecek tahminleri yapması mümkündür. Trend analizi ile hizmetlerdeki mevsimsel etkiler de net bir şekilde görüntülenecektir. Ziyaretlerin işlemlere oranları alınarak farklı analizler yapmak mümkündür.

4) Personel bazlı performans analizi raporu:

En Fazla İşlem Hacmine Sahip Doktor Analizi						
Y-M-D	Ocak					
HealthPlan	SSK					
Column Labels						
Visit Count		Volume		Total Visit Count Total Volume		
Row Labels	Ayaktan Hasta	Yatan Hasta	Ayaktan Hasta	Acil		
Physician	116	194	469	5	116	668
AYŞEGÜL KARAHAN	21		53		21	53
ENİS OĞUZ	35	14	78		35	92
ERTAN ÖKMEN	32	8	63	1	32	72
FATMA İNCİ KÜÇÜKERCAN		19	19			38
GÜLCAN BALOĞLU		25	20	1		46
GÜRSEL ATEŞ	13	23	36		13	59
İ. HAKKI NECDET ÜSKENT	13	3	17		13	20
İNCİ KARAARSLAN		54	6	3		63
METİN GÜDEN	2	1	176		2	177
SERTAÇ ÇİÇEK		47	1			48
Grand Total	116	194	469	5	116	668

Şekil 6.46 Personel bazlı performans analizi raporu

Bu rapor tarih filtresinin yanında sağlık güvencesi filtresini de içermektedir. Analizler bu bazda yapılır. Çalışan performansları ve işlemleri bu rapor aracılığı ile incelenerek ücretlendirme politikaları üretilebilir. Ayrıca eksiklik görülen konularda eğitim planları da yapılabilir.

7. SONUÇ

İş zekası, bir organizasyonun ham verilerini analiz etmek için kullandığı çeşitli yazılım uygulamalarının tümünü kapsayan veri madenciliği, online analitik işleme, sorgulama ve raporlama dahil olmak üzere birbiriyle entegre çalışan çeşitli aktivitelerden oluşan bir disiplindir. Doğru verilerin anlamlı ve işe yarar bilgiler haline dönüştürülmesi, bu bilgiler ile geleceğe dönük tahminler yapılması, bilgi ve tahminler doğrultusunda karar önerilerinin hazırlanması bu teknolojinin işlevleridir.

Uygulama kapsamında farklı veri kaynaklarından elde edilen veriler tasarlanan veri ambarı ve veri kübü üzerinde modellendi. Veri ambarı tasarımı aşamasında tabloların oluşturulması için SQL Server 2008 R2 aracı, veri kaynaklarından veri ambarına data aktarım işlemi için SQL Server 2008 SSIS, çok boyutlu veri modeli tasarım için Microsoft SQL Server ile gelen Business Intelligence Development Studio (BIDS), analiz işlemi için ise SQL Server 2008 SSAS araçları kullanıldı.

Analiz sistemi, veri ambarından elde edilen verileri çok boyutlu, kullanıcının rahatlıkla yönetebileceği küpler haline getirir. OLAP küpleri halindeki verilerin kullanıcıyla buluşması MS Excel 2007 ile sağlandı. Kullanıcı verilere mevcut boyutlardan bakarak normalde elde edilmesi zor, karmaşık hesaplamalara rahatlıkla ulaşır, analizler yaparak karar destek noktasında kendisine yardımcı olacak bilgiler elde eder.

Uygulama kapsamında, teorik olarak değinilen konuların tamamının yer almasına özen gösterildi. İş zekası çözümü insan yaşamında kritik yeri olan sağlık sektörü için tasarlanıp, ihtiyaçlarına uygun bir mimari yaratıldı.

Veri kaynağı ve veri ambarının tasarımı tamamlandıktan sonra gerçekleştirilen aktarım işleminde veri temizleme ve dönüştürme işlemleri uygulandı. Çok boyutlu veri modeli tasarımında veri tabanı şeması olarak yıldız ve kar tanesi şemaları kullanılıp, veri kübü tasarımında kavram hiyerarşilerine yer verilip, analiz aşamasında OLAP operasyonları kullanılmıştır.

İş zekası çözümü kapsamında kurumsal performansın gelişimine yönelik bir çalışma yürütülüp kurumun strateji haritası tasarlandı. Strateji haritasını oluşturmak yönetici ekip için, kuruluşu müşterilerini tanımlayan ve bunu net bir şekilde odaklanmış bir stratejiye dönüştüren bir işletme olarak görme açısından tam bir dönüm noktası oldu.

Bunun neticesinde, tüm sađlık bakım sistemini odaklamak için bir performans yönetimi aracı ortaya çıktı. Aylık karne gözden geçirme toplantıları yapıldı. Karne genel örgütsel performansının kolayca taranıp hazmedilmesini ve gerekli düzenlemelerin yapılmasını sağladı.

Strateji haritasının belirlenmesi ve iş zekası siteminin kurumda uygulamaya alınması ile yönetici ekip büyümeyen alanları destekleyecek büyüme hedefli alanlar belirledi. Çapraz sübvansiyon sayesinde ihtiyaç duyulan fakat az kar getiren hizmetlerini hastaları için elde turmayı başardı.

Ayrıca kurum müşterilerinin segmentasyonunu üç farklı gruba ayırıp her grup için doğru deđer önermesinin belirlenmesini başardı. Primer bakım hastaları için müşteri samimiyet stratejisi gerektiđi belirlendi. Bu hastaların kuruma geldiklerinde ya da aradıklarında tüm sađlık geçmişlerini baştan sona anlatmak zorunda kalmayacaklarını bilmeye ihtiyaçları vardır. Özel bakıma ihtiyacı olan hastalar, bu grup kusursuz klinik ortamına, gelişmiş teknolojiye ve uzmanlığa büyük önem verirler. Kurum bu grup için ürün liderliği stratejisini uygular. Müşteri gruplarından sonuncusu ödeme yapanlar, yani kurumdan hizmet satın alanlardır. Bu grup, düşük maliyetli hizmet ve yenilikçi sađlık-bakım maliyeti teminatı programları talep ederler. Çalışanlarına ve müşterilerine paraları karşılığında en iyi deđerı önermek isterler. Bu da düşük toplam maliyet stratejisi demektir.

İş zekası sistemine geçildikten sonra karlılıkta yüksek oranda artış, ilaç ve maaş maliyetlerine rağmen taburcu edilen hasta başına maliyet ile kabul edilen hasta başına maliyetin dengelenmesi, kliniklerin alacak hesaplarında on günlük bir azalma, primer bakım kliniklerinde randevu erişimi bulunabilirliğinde %12'lik bir iyileşme, genel hasta memnuniyetinde %15'lik bir iyileşme, genel klinik hastası memnuniyetinde %11'lik bir iyileşme ölçümlenmiştir.

Çalışmanın bir sonraki safhası ise hazırlanan veri ambarı yapısının kaynaklık edeceği bir veri madenciliđi projesinin hayata geçirilmesidir. Artificial intelligence, fuzzy logic, machine learning, neural networks gibi disiplinler ile bu uygulama hayata geçirilebilir.

KAYNAKLAR

Almeida, M. S., Ishikawa, M., Reinschmidt, J. ve Roeber, T., (1999), Getting Started with Data Warehouse and Business Intelligence, IBM, California.

Anahory, S. ve Murray, D., (1997), Data Warehousing in The Real World: A Practical Guide for Building Decision Support Systems, Addison-Wesley, Boston.

Andersson, D., Fries, H. ve Johansson, P., (2008), “Business Intelligence: The Impact On Decision Support and Decision Making Processes”, Lisans Tezi, Jönköping International Business School, Sweden.

Azvine, B., Cui, Z. ve Nauck, D. D., (2005), “Towards Real-Time Business Intelligence”, BT Technology Journal, 23(3):214-225.

Bain, T., Benkovich, M., Dewson, R., Ferguson, S., Graves, C., Joubert, T., Lee, D., Scott, M., Skoglund, R., Turley, P., Youness, S., (2001), Professional SQL Server 2000 Data Warehousing with Analysis Services, Wrox Press, Birmingham.

Biere, M., (2003), Business Intelligence For The Enterprise, Prentice Hall PTR, New Jersey.

Boakye, J. K., (1999), “An Empirical Assessment Of Data Warehousing Implementation Effectiveness”, Doktora Tezi, Walden University.

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

Colin, J., (1997), “The Future of The Organization: Selected Knowledge Management Issues”, Journal of Knowledge Management, 1(1):15-26.

Çağıltay, N. E., (2010), İş Zekası ve Veri Ambarı Sistemleri, ODTU Yayıncılık, Ankara.

Datta, A. ve Thomas, H., (1999), “The Cube Data Model: A Conceptual Model and Algebra for Online Analytical Processing in Data Warehouses”, Decision Support Systems, 27(3):289-301.

Gray, P. ve Watson, H. J., (1998), Decision Support in The Data Warehouse, Prentice Hall PTR, New Jersey.

Han, J. ve Kamber, M., (2006), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, USA.

Howson, C., (2008), Successful Business Intelligence – Secrets to Making BI a Killer App., The McGraw Hill, New York.

Inmon, W. H. ve Hackathorn, R. D., (1994), Using The Datawarehouse, Wiley, New York.

Kash, R., (2002), *The New Law of Demand and Supply: The Revolutionary New Demand Strategy for Faster Growth and Higher Profits*, Doubleday Business, New York.

Kaplan R. S. ve Norton D. P., (2006), *Strateji Haritaları Gayrimaddi Varlıkları Maddi Sonuçlara Dönüştürmek*, Alfa Basım, İstanbul.

Kelly, S., (1996), *Data Warehousing: The Route To Mass Customisation*, John Wiley & Sons, UK.

Kimball, R. ve Ross, M., (2002), *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)*, John Wiley & Sons, UK.

Koutsoukis, N. S., Mitra, G. ve Lucas, C., (1999), "Adapting Online Analytical Processing for Decision Modelling: The Interaction of Information and Decision Technologies", *Decision Support Systems*, (26):1-30.

Kramer, G. ve Nisbet, S., (2008), "The Practical Union of OLAP Analysis and Geographic Mapping", SAS, North Carolina.

March, S. T. ve Hevner, A. R., (2005), "Integrated Decision Support Systems: A Data Warehousing Perspective", *Decision Support Systems*, 43(3):1031-1043.

Mattison, R., (1996), *Data Warehousing: Strategies, Technologies and Techniques*, McGraw Hill, New York.

Moss, L. T. ve Atre, S., (2003), *Business Intelligence Roadmap: The Complete Project Lifecycle For Decision - Support Applications*, Addison - Wesley, Boston.

Nagabhushana, S., (2006), *Data Warehousing OLAP and Data Mining*, New Age International Publishers, USA.

Reinschmidt, J. ve Francoise, A., (2000), *Business Intelligence Certification Guide*, IBM, California.

Roddick J. F., Fule P. ve Graco W. J., (2003), "Exploratory Medical Knowledge Discovery: Experience and Issues", *ACM SIGMOD Record*, 5(1):94-95.

Shin, B., (2002), "A Case of Data Warehousing Project Management", *Information and Management*, 39(7):581-592.

Stackowiak, R., Rayman, J. ve Greenwald, R., (2007), *Oracle Data Warehousing and Business Intelligence Solutions*, Wiley, New York.

Turban, E., Aronson J. E., Liang T. P. ve Sharda R., (2006), *Decision Support Systems and Intelligent Systems 8th Edition*, Prentice Hall, New Jersey.

Utle, C., (2008), *Business Intelligence with Microsoft Office Performance Point Server 2007*, The McGraw Hill, New York

Vassiliadis, P., (2000), Data Warehouse Modeling and Quality Issues, Doktora Tezi, National Technical University of Athens, Greece.

Watson, H. J., Goodhue, D. L. ve Wixom, B. H., (2002), "The Benefits of Data Warehousing: Why Some Organizations Realize Exceptional Payoffs", Information and Management, 39(6):491-502.

Zeng, Y., Roger, H. L. ve Yen, C., (2003), "Enterprise Integration with Advanced Information Technologies: ERP and Data Warehousing", Information Management & Computer Security, 11(3):115-122.

İnternet Kaynakları

- [1] http://www.cio-club.net/Makaleler/PDF/Dosya_isZekasi_Nisan.pdf
- [2] http://www.germany.businessobjects.com/pdf/smallbusiness/wp_bi_for_it.pdf
- [3] <http://www.sap.com/platform/netweaver/index.epx>
- [4] <http://www.microstrategy.com/bi-applications/industries/healthcare.asp>
- [5] <http://www.datawarehouse.gen.tr/index.php/component/k2/itemlist/tag/etl>
- [6] <http://programmatik.info/Others.aspx>
- [7] http://gkmc.utah.edu/ebis_class/2003s/Oracle/Cookbook/dm_conce.htm
- [8] http://www.olap.com/w/index.php/Category:OLAP_History
- [9] http://en.wikipedia.org/wiki/Online_analytical_processing#cite_note-15
- [10] http://www.symcorp.com/downloads/OLAP_CouncilWhitePaper.pdf
- [11] <http://www.databasejournal.com/sqlc/article.php/1457031/OLAP-and-DataWarehousing-The-Problem-and-Solution.htm>
- [12] http://download.oracle.com/docs/html/B13915_04/images/hierarchy1.gif
- [13] http://download.oracle.com/docs/html/B13915_04/images/cube2.gif
- [14] http://wappreview.de/tino/projekt_handel_III/datawrhs/slice.htm
- [15] <http://www.directionsmag.com/articles/creating-and-manipulating-multidimensional-tables-with-locational-data-usin/123565#>
- [16] <http://www.developpez.net/forums/d86373/logiciels/solutions-dentreprise/business-intelligence/rolap-molap-holap-distinguer-utiliser/>

EKLER

Ek 1 Tabloların Oluřturulma Detayları.....	89
Ek 2 Kurumun Strateji Haritası.....	93

Ek 1 Tabloların Oluşturulma Detayları

```

CREATE TABLE FacilityDim
(
    FACILITYID int,
    NAME nvarchar(255),
    SHORTNAME nvarchar(255),
    ISEXTERNAL nvarchar(255),
    ACTIVE nvarchar(255)
)

CREATE TABLE HealthPlanTypeDim
(
    HEALTHPLANTYPEID int,
    NAME nvarchar(255)
)

CREATE TABLE HealthPlanPayerDim
(
    HEALTHPLANPAYERID int,
    HEALTHPLANTYPEID int,
    NAME nvarchar(255),
    CODE nvarchar(255),
    DESCRIPTION nvarchar(255),
    ACTIVE nvarchar(255)
)

CREATE TABLE OrgSubTypeDim
(
    ORGUNITSUBTYPEID nvarchar(255),
    ORGUNITTYPEID nvarchar(255),
    NAME nvarchar(255)
)

CREATE TABLE OrgTypeDim
(
    ORGUNITTYPEID nvarchar(255),
    NAME nvarchar(255),
    PARENTID nvarchar(255),
    ACTIVE nvarchar(255)
)

CREATE TABLE ResourceDim
(
    RESOURCEID int,
    NAME nvarchar(255),
    DEPTID nvarchar(255),
    LOCATIONID nvarchar(255),
    RESOURCETYPEID nvarchar(255)
)

CREATE TABLE dbo.ResourceTypeDim
(

```

```

        RESOURCETYPEID nvarchar(255),
        NAME nvarchar(255),
        PARENTID nvarchar(255),
        ACTIVE nvarchar(255),
        ISPROVIDER nvarchar(255)
    )

CREATE TABLE ServiceDim
(
    SERVICEID int,
    NAME nvarchar(255),
    DEPTID int,
    SERVICESUBTYPEID nvarchar(255),
    SERVICETYPEID nvarchar(255),
    SERVICECENTERID int
)

CREATE TABLE VisitTypeDim
(
    VISITTYPEID int,
    NAME nvarchar 255
)

CREATE TABLE dbo.DateDim(
    DateKey int,
    Date datetime,
    DateName nvarchar(50),
    DateNameTR nvarchar(50),
    YearKey int,
    Year datetime,
    YearName nvarchar(50),
    YearNameTR nvarchar(50),
    HalfYearKey int,
    HalfYear datetime,
    HalfYearName nvarchar(50),
    HalfYearNameTR nvarchar(50),
    QuarterKey int,
    Quarter datetime,
    QuarterName nvarchar(50),
    QuarterNameTR nvarchar(50),
    MonthKey int,
    Month datetime,
    MonthName nvarchar(50),
    MonthNameTR nvarchar(50),
    WeekKey int,
    Week datetime,
    WeekName nvarchar(50),
    WeekNameTR nvarchar(50),
    IsHoliday bit,
    HolidayKey int,
    IsSpecialDay bit,
    SpecialDayKey int,
    IsWorkDay bit,
    DayOfYear int,
    DayOfYearName nvarchar(50),
    DayOfYearNameTR nvarchar(50),
    DayOfHalfYear int,

```

```

DayOfHalfYearName nvarchar(50),
DayOfHalfYearNameTR nvarchar(50),
DayOfQuarter int,
DayOfQuarterName nvarchar(50),
DayOfQuarterNameTR nvarchar(50),
DayOfMonth int,
DayOfMonthName nvarchar(50),
DayOfMonthNameTR nvarchar(50),
DayOfWeek int,
DayOfWeekName nvarchar(50),
DayOfWeekNameTR nvarchar(50),
WeekOfYear int,
WeekOfYearName nvarchar(50),
WeekOfYearNameTR nvarchar(50),
MonthOfYear int,
MonthOfYearName nvarchar(50),
MonthOfYearNameTR nvarchar(50),
MonthOfHalfYear int,
MonthOfHalfYearName nvarchar(50),
MonthOfHalfYearNameTR nvarchar(50),
MonthOfQuarter int,
MonthOfQuarterName nvarchar(50),
MonthOfQuarterNameTR nvarchar(50),
QuarterOfYear int,
QuarterOfYearName nvarchar(50),
QuarterOfYearNameTR nvarchar(50),
QuarterOfHalfYear int,
QuarterOfHalfYearName nvarchar(50),
QuarterOfHalfYearNameTR nvarchar(50),
HalfYearOfYear int,
HalfYearOfYearName nvarchar(50),
HalfYearOfYearNameTR nvarchar(50)
)

CREATE TABLE VisitFact
(
    IVID int,
    IPID int,
    VISITTYPEID int,
    ORGUNITID int,
    STARTDATEKEY int,
    ENDDATEKEY int
)

CREATE VIEW [dbo].[VolumeFact] AS
SELECT
    CONVERT(INT,VA.SERVICEID) AS SERVICEID,
    CONVERT(INT,RESOURCEID) AS RESOURCEID ,
    CONVERT(INT, REPLACE(IPID, '.0000', '')) AS IPID ,
    CONVERT(INT, IVID) AS IVID , CONVERT(INT, ORDERID) AS ORDERID ,
    CONVERT(INT, REPLACE(VISITTYPEID, '.0000', '')) AS VISITTYPEID,
    CONVERT(INT, CONVERT(NVARCHAR(8), STARTDATE, 112)) AS DATEKEY,
    CONVERT(INT, VA.SERVICECENTERID) AS SERVICECENTERID,
    CONVERT(INT, FOS.FACILITYID) AS FACILITYID, 1 AS DataTypeId
FROM volume_operations_02a VA (NOLOCK), FacilityOrganizationServiceDim
FOS
WHERE VA.SERVICEID = FOS.SERVICEID

```

UNION ALL

SELECT

```

    CONVERT (INT, VB.SERVICEID) ,
    CONVERT (INT, SURGEON) as RESOURCEID,
    CONVERT (INT, REPLACE (IPID, '.0000', '')) ,
    CONVERT (INT, IVID) , CONVERT (INT, ORDERID) ,
    CONVERT (INT, REPLACE (VISITTYPEID, '.0000', '')) ,
    CONVERT (INT, CONVERT (NVARCHAR (8), STARTDATE, 112)) AS DATEKEY,
    CONVERT (INT, VB.ORGUNITID) AS SERVICECENTERID,
    CONVERT (INT, FOS.FACILITYID), 1 AS DataTypeId
FROM volume_operations_02b VB, FacilityOrganizationServiceDim FOS
WHERE VB.SERVICEID = FOS.SERVICEID

```

UNION ALL

SELECT

```

    CONVERT (INT, VD.SERVICEID) ,
    CONVERT (INT, RESOURCEID) ,
    CONVERT (INT, REPLACE (IPID, '.0000', '')) ,
    CONVERT (INT, IVID) , CONVERT (INT, ORDERID) ,
    CONVERT (INT, REPLACE (VISITTYPEID, '.0000', '')) ,
    CONVERT (INT, CONVERT (NVARCHAR (8), STARTDATE, 112)) AS DATEKEY,
    CONVERT (INT, FOS.SERVICECENTERID) ,
    CONVERT (INT, FOS.FACILITYID), 2 AS DataTypeId
FROM volume_controls_02d VD, FacilityOrganizationServiceDim FOS
WHERE VD.SERVICEID = FOS.SERVICEID

```

UNION ALL

SELECT

```

    CONVERT (INT, VE.SERVICEID) ,
    CONVERT (INT, RESOURCEID) ,
    CONVERT (INT, REPLACE (IPID, '.0000', '')) ,
    CONVERT (INT, IVID) , CONVERT (INT, ORDERID) ,
    CONVERT (INT, REPLACE (VISITTYPEID, '.0000', '')) ,
    CONVERT (INT, CONVERT (NVARCHAR (8), STARTDATE, 112)) AS DATEKEY,
    CONVERT (INT, FOS.SERVICECENTERID) ,
    CONVERT (INT, FOS.FACILITYID), 3 AS DataTypeId
FROM volume_muayene_02e VE, FacilityOrganizationServiceDim FOS
WHERE VE.SERVICEID = FOS.SERVICEID

```

UNION ALL

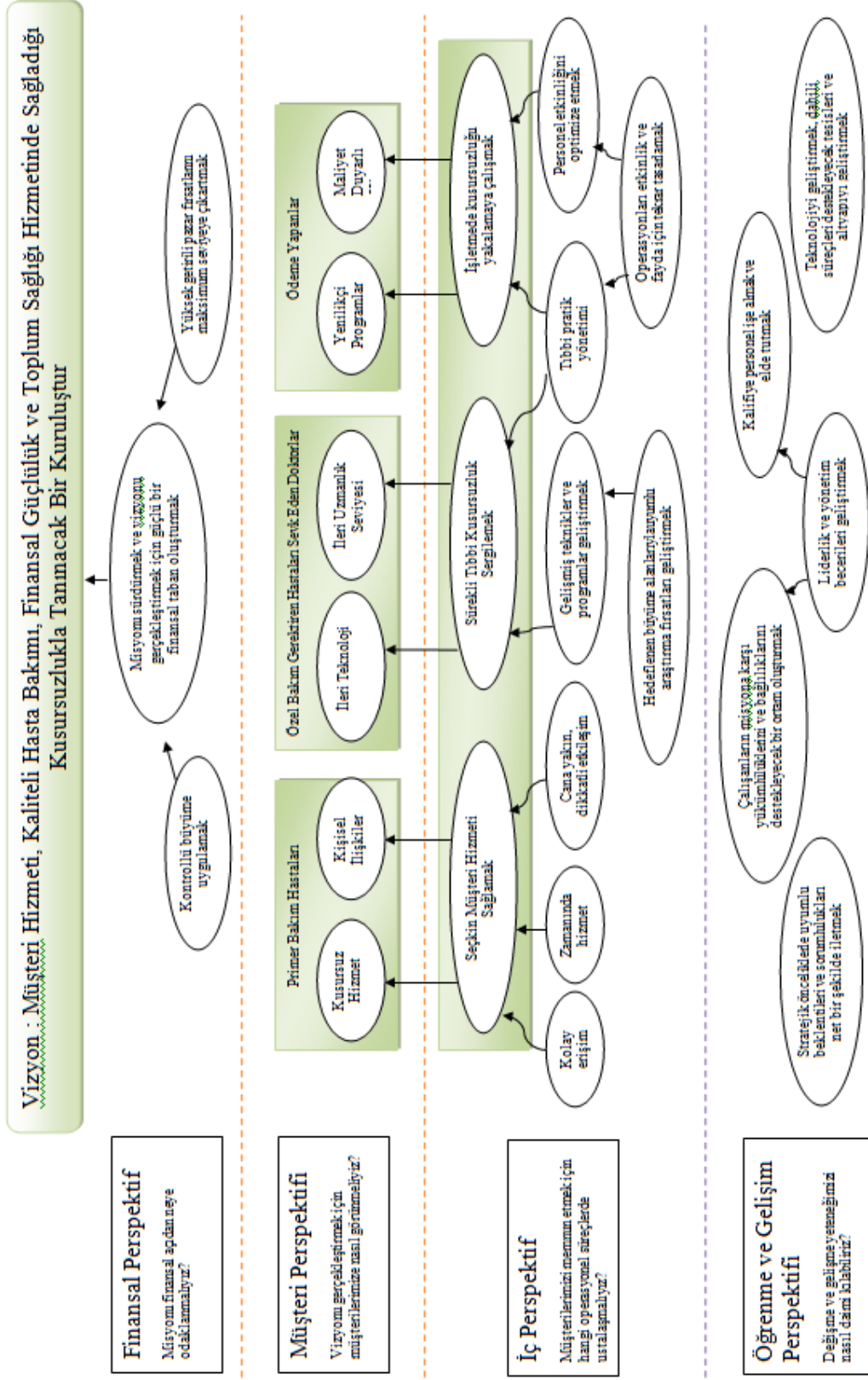
SELECT

```

    CONVERT (INT, VG.SERVICEID) AS SERVICEID,
    CONVERT (INT, RESOURCEID) AS RESOURCEID,
    CONVERT (INT, REPLACE (IPID, '.0000', '')) AS IPID,
    CONVERT (INT, IVID) AS IVID, CONVERT (INT, ORDERID) AS ORDERID,
    CONVERT (INT, REPLACE (VISITTYPEID, '.0000', '')) AS VISITTYPEID,
    CONVERT (INT, CONVERT (NVARCHAR (8), STARTDATE, 112)) AS DATEKEY,
    CONVERT (INT, FO.SERVICECENTERID) AS SERVICECENTERID,
    CONVERT (INT, VG.FACILITYID) AS FACILITYID, 4 AS DataTypeId
FROM volume_non_svc_search_services_02g
VG, FacilityOrganizationServiceDim FO WHERE VG.SERVICEID = FO.SERVICEID

```

Ek 2 Kurumun Strateji Haritası



ÖZGEÇMİŞ

Doğum tarihi	15.10.1983	
Doğum yeri	Sivas	
Lise	1998-2002	Şehremini Lisesi
Lisans	2002-2007	Yıldız Üniversitesi Kimya-Metalurji Fak. Matematik Mühendisliği Bölümü
Yüksek Lisans	2007-2011	Yıldız Teknik Üniversitesi Fen Bilimleri Enst. Endüstri Müh. Anabilim Dalı, Endüstri Müh.

Çalıştığı kurumlar

2006-2008	Akademi Yazılım A.Ş.
2008-2010	Axa Sigorta A.Ş.
2010-	Akbank T.A.Ş