

**HASTANE BİLGİ YÖNETİM SİSTEMLERİNDE OLAP
YÖNTEMLERİYLE KARAR DESTEK MODÜLÜ GELİŞTİRMEK**

**Pamukkale Üniversitesi
Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı**

Murat TAŞER

Danışman: Yrd. Doç. Dr. A. Kadir YALDIR

**Mayıs 2008
DENİZLİ**

YÜKSEK LİSANS TEZİ ONAY FORMU

Murat TAŞER tarafından Yrd. Doç. Dr. A. Kadir YALDIR yönetiminde hazırlanan “**Hastane Bilgi Yönetim Sistemlerinde Olap Yöntemleriyle Karar Destek Modülü Geliştirmek**” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans Tezi olarak kabul edilmiştir.



Doç. Dr. Fahir DEMİRKAN
Jüri Başkanı



Yrd. Doç. Dr. Gürhan GÜNDÜZ
Jüri Üyesi



Yrd. Doç. Dr. A. Kadir YALDIR
Jüri Üyesi (Danışman)

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Mehmet Ali SARIGÖL
Müdür

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etiĐe ve akademik kurallara özenle riayet edildiĐini; bu alıřmanın doĐrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etiĐe uygun olarak kaynak gösterildiĐini ve alıntı yapılan alıřmalara atfedildiĐini beyan ederim.

İmza:

ÖĐrenci Adı Soyadı: Murat TAŐER

TEŐEKKÜR

Öneri, destek ve yardımlarından dolayı değerli danışman hocam Yrd. Doç. Dr. A. Kadir YALDIR, çalışma arkadaşlarım Sayın Ayşegül ŐAVLI ve Mustafa BAYRAK'a, tez çalışmasını Bilimsel Araştırma Projesi kapsamında destekleyen Pamukkale Üniversitesi B.A.P. birimine, son olarak tez çalışmalarım sırasında benden manevi desteğini esirgemeyen Annem Bilge TAŐER ve Eşim Eda TAŐER'e teşekkürü borç bilirim.

ÖZET

HASTANE BİLGİ YÖNETİM SİSTEMLERİNDE OLAP YÖNTEMLERİYLE KARAR DESTEK MODÜLÜ GELİŞTİRMEK

Taşer, Murat
Yüksek Lisans Tezi, Bilgisayar Mühendisliği ABD
Tez Yöneticisi: Yrd. Doç. Dr. A. Kadir YALDIR
Mayıs 2008, 92 Sayfa

Günümüzde sağlık sektörü, verilen hizmetten dolayı hastanın memnuniyeti, giderler ve kazanımlar, tanı ve tedaviye yönelik tıbbi araştırmalar gibi konularda karar vermeye yönelik bilgi açlığı çekmektedir.

Çalışmada, bu ihtiyaca cevap vermek amacıyla, niteliksiz veri bombardımanı altında ne yapacağını şaşırarak sağlık sektörü yöneticilerine, zaman dilimli ve çok boyutlu veri görünümünü sağlayan OLAP analiz işlemi ışığında yol gösterilmeye çalışılmıştır. Pamukkale Üniversitesi Eğitim Uygulama ve Araştırma Hastaneleri yöneticilerinin de fikirlerinin alınması sonucu, 16 değişik karar destek noktasında veri küpleri oluşturulmuştur. Kullanıcıların küplere kolayca ulaşması, üzerinde çalışabilmesi ve bu küpler sayesinde doğru ve öznet bilgiye hızlıca ulaşmaları amacıyla bir uygulama geliştirilmiştir.

OLAPPROG adını verdiğimiz bu uygulama, kullanıcılarının dice, slice, drill up, drill down gibi OLAP özelliklerinden yararlanmasını sağlamakla birlikte, küp görüntüsünün özelleştirilmesi, grafik çıktısının alınması ve farklı dosya formatları ile kaydedilmesi işlemlerini de yapabilmelerini sağlar.

Veri transferi, zamanlanmış DTS paketleriyle yapıldığı için ve geliştirilen uygulama eklenen her yeni veri küpünü Microsoft Analysis Services'e bağlanarak otomatik olarak tanıdığından çalışmanın sürekliliği de sağlanmıştır.

Çalışmada, Microsoft SQL Server 2000'in Database Server, SQL Agent, Data Transformation Services ve Analysis Services bileşenleri kullanılmıştır. OLAPPROG uygulaması ise Borland Delphi 7.0 yazılım geliştirme ortamında kodlanmıştır.

Anahtar Kelimeler: Hastane Bilgi Yönetim Sistemi, OLAP, Karar Destek Sistemleri, KDS, Data Transformation Services, DTS, Microsoft Analysis Services

Doç. Dr. Fahir DEMİRKAN
Yrd. Doç. Dr. A. Kadir YALDIR
Yrd. Doç. Dr. Gürhan GÜNDÜZ

ABSTRACT**DEVELOPING DECISION SUPPORT MODULE BY USING OLAP METHODS
IN HOSPITAL INFORMATION SYSTEMS**

Taşer, Murat
M. Sc. Thesis in Computer Engineering
Supervisor: Asst. Prof. Dr. A. Kadir YALDIR
May 2008, 92 Pages

Health sector in nowadays is in lack of information on subjects such as patient satisfaction on given services, expenses and incomes, and medical studies on diagnosis and treatment.

In this study, it's aimed to guide the health sector administrators, who are confused under featureless data bombardment, by using OLAP analysis that provides time segment and multidimensional data view in order to prevent this need. By getting the opinions of Pamukkale University Education, Practice and Research Hospital administrators, data cubes on 16 different decision support points are formed. An application is developed in order to provide users to reach the cubes easily, to work on them and to reach correct information and summarise the information quickly with the help of these cubes.

The application we named OLAPPROG provides the users to customize the cube view, to get the printout of the graphic and to save with different file formats in addition to benefit from the OLAP features such as dice, slice, drill up and drill down.

Continuity of this study is also provided through the fact that data transfer is done by timed DTS packets and application recognizes each attached data cubes automatically by connecting Microsoft Analysis Services.

In the study, Database Server, SQL Agent, Data Transformation Services and Analysis Services components of Microsoft SQL Server 2000 are used. OLAPPROG application is coded under Borland Delphi 7.0 software development package.

Keywords: Hospital Information Systems, OLAP, Decision Support Systems, DSS, Data Transformation Services, DTS, Microsoft Analysis Services

Assoc. Prof. Dr. Fahir DEMİRKAN
Asst. Prof. Dr. A. Kadir YALDIR
Asst. Prof. Dr. Gürhan GÜNDÜZ

İÇİNDEKİLER

	<u>Sayfa No</u>
1. GİRİŞ	1
2. KDS VE OLAP.....	3
2.1 Karar Destek Sistemleri (KDS)	3
2.2 Online Analytical Processing(OLAP).....	5
2.2.1 OLAP özellikleri	6
2.2.2 OLAP’ın avantajları	9
3. HASTANE BİLGİ YÖNETİM SİSTEMİ (HBYS).....	10
3.1 HBYS’nin Yararları	11
3.2 Genel HBYS Modülleri.....	11
4. KARAR DESTEK MODÜLÜ VERİ AMBARININ OLUŞTURULMASI.....	13
4.1 Veri Temizleme İşlemi	13
4.2 Karar Destek Modülü Raporları.....	14
4.2.1 Hasta kimlik bilgileri raporu	15
4.2.2 Paket hastaların bölüme göre dağılım raporu	16
4.2.3 Pakete dönüştürülen referansların hizmet türüne göre dağılım raporu	18
4.2.4 Paket ameliyatların bölüme göre dağılım raporu	19
4.2.5 Hastaların ayaktan/yatan durumuna ve bölümüne göre dağılım raporu.....	20
4.2.6 Faturaların kuruma göre dağılım raporu.....	22
4.2.7 Yatan hastaların tanı ve ABD’na göre dağılım raporu	23
4.2.8 Ayaktan hastaların tanı ve ABD’na göre dağılım raporu	25
4.2.9 Tanı hasta dağılım raporu	26
4.2.10 Hizmet türüne ve bölüme göre dağılım raporu	27
4.2.11 Tedavisi tamamlanmış hastaların uzman hekim onayı alınmamış radyoloji, laboratuvar, patoloji işlemlerinin dağılım raporu	28
4.2.12 Hastaların sosyal güvencelerine göre fatura değerlerinin dağılım raporu ...	30
4.2.13 Uzman doktor/hasta türü katkı payı dağılım raporu.....	31
4.2.14 Hasta makbuz bilgileri dağılım raporu	32
4.2.15 Vaka başı hastaların bölüme göre dağılım raporu.....	34
4.2.16 Bölüme göre öğretim üyeleri performans uygulaması analizi	35
4.3 DTS Paketlerinin Oluşturulması	37
4.4 DTS Paketlerinin Zamanlanması	40
5. ANALİZ VERİTABANININ OLUŞTURULMASI	43
5.1 Analysis Services Öğeleri.....	44
5.1.1 Analysis server	44
5.1.2 Analysis manager	44
5.1.3 Analiz veritabanı	45
5.1.4 Pivot table service	45
5.2 Analiz Veritabanı Öğeleri.....	46
5.2.1 Data source.....	46
5.2.2 Cube (Küp).....	47
5.3 OLAP Küpü Oluşturmak	47
5.4 Process.....	56

6. OLAPPROG YAZILIMI.....	60
6.1 Kp Grnts Seimlerinin Yapılması	61
6.1.1 Kp bilgisi ve kpe baėlanma.....	61
6.1.2 Dimension layout	62
6.1.3 Measure ve properties.....	62
6.2 Kp Grnts zerinde alıřma	64
6.2.1 Sorting.....	64
6.2.2 Filtering.....	64
6.2.3 Grouping	67
6.3 Grafik Desteėi.....	67
6.4 Dosya Olarak Kaydetme-ıktı Alma	69
6.5 Ayarlar	70
6.6 OLAPPROG'un Farklı HBYS'lerde Uygulanabilirliėi	72
7. SONULAR.....	73
KAYNAKLAR.....	74
EKLER.....	76
ZGEMİř.....	92

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 2.1 Karar Destek Sistemleri'nin temel bileşenleri (Çetinyokuş ve Gökçen 2002) ..	5
Şekil 2.2 SQL Server Data Warehousing (WEB_2 2007).....	8
Şekil 4.1 SQL Server Enterprise Manager Wizard ekranı.....	38
Şekil 4.2 SQL Server Enterprise Manager DTS veri taşıma yöntemi.....	38
Şekil 4.3 SQL Server Enterprise Manager DTS sorgu girme ekranı.....	39
Şekil 4.4 DTS paketinin kaydedilmesi ve zamanlanması ekranı.....	40
Şekil 4.5 DTS Wizard zamanlama ekranı.....	41
Şekil 4.6 DTS Packages.....	41
Şekil 4.7 SQL Server Agent Jobs listesi.....	42
Şekil 4.8 Oluşturulan bir DTS paketinin görüntüsü.....	42
Şekil 5.1 Analysis Services Architecture (Microsoft Corporation 2000).....	43
Şekil 5.2 OLAP depolama şemaları (Microsoft Corporation 2000).....	44
Şekil 5.3 Çalışmadaki Analysis Manager öğeleri.....	45
Şekil 5.4 Pivot Table Service (Microsoft Corporation 2000).....	46
Şekil 5.5 Data Source tanımı.....	46
Şekil 5.6 Analysis Manager New Cube Wizard.....	47
Şekil 5.7 New Cube Wizard Fact Table seçimi.....	48
Şekil 5.8 New Cube Wizard Measure seçimi.....	49
Şekil 5.9 Dimension Table seçim ekranı.....	50
Şekil 5.10 Star schema (Microsoft Corporation 2000).....	51
Şekil 5.11 Snowflake schema (Microsoft Corporation 2000).....	51
Şekil 5.12 Dimension hiyerarşi seçim ekranı.....	52
Şekil 5.13 Dimension tipi seçim ekranı.....	53
Şekil 5.14 Standart Dimension Level seçimi.....	53
Şekil 5.15 Time Dimension seçimi.....	54
Şekil 5.16 Time Dimension Level seçimi.....	55
Şekil 5.17 Dimension sihirbazı tamamlama ekranı.....	55
Şekil 5.18 Cube Editör.....	56
Şekil 5.19 Process metodları.....	57
Şekil 5.20 Process işlemi.....	58
Şekil 5.21 Hospital Analiz Veritabanı Process DTS paketi.....	58
Şekil 6.1 OLAPPROG yazılımı.....	61
Şekil 6.2 OLP_CUBES tablosundan görünüm.....	62
Şekil 6.3 Dimension yerleşimi.....	63
Şekil 6.4 Measure seçimi.....	64
Şekil 6.5 Sorting.....	65
Şekil 6.6 Hierarchy Editor.....	66
Şekil 6.7 Member filtreleme.....	66
Şekil 6.8 Member gruplama.....	67
Şekil 6.9 Gruba Member atama.....	68
Şekil 6.10 Grafik –tek Dimension ve tek Measure.....	68
Şekil 6.11 Yoğun verili ve iki Dimension'lı grafik görünümü.....	69
Şekil 6.12 OLAPPROG Excel belgesi.....	70
Şekil 6.13 Görsel ayarlar ekranı.....	70
Şekil 6.14 Veri alanı ayarları – tümünü göster.....	71
Şekil 6.15 Proje geliştirme adımları.....	72

SİMGE VE KISALTMALAR DİZİNİ

ABD	Ana Bilim Dalı
BUT	Bütçe Uygulama Talimatnamesi
DSS	Decision Support System
DTS	Data Transformation Services
ETL	Extract Transform Load
HBYS	Hastane Bilgi Yönetim Sistemleri
KDS	Karar Destek Sistemleri
OLAP	Online Analytical Processing
OLTP	On-Line Transaction Processing
PACS	Picture Archiving and Communications System
PTS	Pivot Table Service
SQL	Structured Query Language

1. GİRİŞ

Günümüzde teknoloji işletmelerin yürütülmesinde ve yönetilmesinde giderek ön plana çıkmaktadır. Hastaneler için vazgeçilmez hale gelen Hastane Bilgi Yönetim Sistemleri (HBYS) bunun bir örneğidir. HBYS'ler iş akışlarına göre birimler arası iletişimlerini sağlayarak ve çalışanların bilgi ihtiyaçlarını karşılayarak, hastane içerisindeki bilgi akışını yöneten yazılım ve donanım bütünü olarak düşünülebilir. Karar Destek Sistemleri (KDS) ise yöneticilere karar verme noktasında yardımcı olan yazılımlardır.

Sürekli artan bir ivmeyle gelişen sağlık sektöründe, hastane yöneticilerinin, verileri değerlendirme ve uygun kararı verme noktasında, bilgiye değişik boyutlardan hızlıca erişim ihtiyacı artmaktadır. Yöneticiler değişik faturalama prosedürleri, birbirinden farklı maliyet hesaplamaları, değişken geri ödeme ve performans değerleri karşısında istatistiki bilgi bombardımanı altında doğru karara ulaşamamaktadır.

Çalışmada bu bilgiye ulaşmadaki sıkıntıları çözmek amacıyla, Pamukkale Üniversitesi Hastaneleri Hastane Bilgi Yönetim Sistemi (HBYS) üzerinde OLAP yöntemi kullanan bir Karar Destek Modülü oluşturulmuştur. Modül, kritik karar destek noktaları olarak belirlenen faturalama, geri ödemeler, doktor ve bölüm performansları, hakedişler, paket ve vakabaşı uygulamaları, hasta kimlik ve tanı bilgileri konularında OLAP tabanlı raporlar içermektedir.

Tez çalışması için yapılan literatür araştırmalarında örnekleri incelenen HBYS Karar Destek Modüllerinin, kullanıcıların belli parametre aralıkları girerek sorguladığı raporlar ve istatistiki bilgilerden oluştuğu görülmüş, Online Analytical Processing (OLAP) yazılım sistemi tabanlı karar destek modülüne sahip HBYS örneğine rastlanmamıştır. Kullanıcılar OLAP teknolojisi ile bilgiye değişik boyutlardan hızlı, zaman yönelimli ve Structured Query Language (SQL) bilgisine gerek duymaksızın erişebilirler.

Çalışma örneğın vaka başı uygulamalarda normal raporlama modüllerinden farklı olarak; seçilen tarih aralığındaki bölümlerin sadece vaka başı fatura değeri ve normal faturalama değerlerini karşılaştırmak yerine, hangi bölümde, hangi tarih aralıklarında ve hangi hasta türünde vaka başı değerlerinin fatura değerlerinden iyi ve hangi durumda kötü olduğunu tek ekranda tekrar tekrar parametre seçmeden hızlı bir şekilde kullanıcıya sunar. Bu sayede değişik boyutlardan zaman yönelimli değerlendirme hızlıca yapılabilir.

İkinci ve üçüncü bölümde, HBYS, KDS ve OLAP kavramları açıklanmış, bu konuda yapılan çalışmaların yararları hakkında bilgiler verilmiştir.

Dördüncü bölümde, ihtiyaç duyulan karar destek noktaları hastane yöneticileri ile görüşülerek belirlenmiş ve bu noktalarda ulaşılmak istenen raporlar çalışmaya eklenmiştir. Bu aşamada doğru ve net bilgiye ulaşabilmek amacıyla, kullanıcı hatası ve program eksikliklerinden kaynaklanan hatalı ve eksik veri girişleri SQL ifadeleri yardımıyla temizlenmiş, istenen özet veri ayrıştırılmış ve Microsoft SQL Server Data Transformation Services (DTS) yardımıyla veri ambarına taşınmıştır. Bu işlemler birer paket haline getirilmiş ve zamanlanarak bilgiye ulaşımın sürekliliği sağlanmıştır.

Beşinci bölümde, elde edilen veriler üzerinde OLAP tabanlı analiz işlemi yapılabilmesi için OLAP küp yapısının, dimension ve measure öğelerinin Microsoft Analysis Services yardımıyla oluşturulması ve küplerin analize hazır hale getirilmesi anlatılmıştır.

Altıncı bölümde, analiz işlemi için geliştirilen ve kullanıcıların rahatça ulaşabilmesi amacıyla HBYS'ye karar destek modülü olarak entegre edilen OLAPPROG yazılımı ile ilgili ayrıntılar ele alınmıştır. Radar-Soft firmasının yazılım araçlarından yararlanılarak oluşturulan OLAPPROG, filtreleme ve gruplama yeteneklerinin yanı sıra, kullanıcıların OLAP küplerine daha kolay, esnek, çıktı ve grafik destekli olarak ulaşmasını, görsel ve veri alanı ayarları yardımıyla küp görüntüsünü özelleştirebilmesini sağlamıştır.

Yedinci bölümde bu çalışma ile elde edilen sonuçlara yer verilmiş olup, çalışmanın devamı niteliği taşıyabilecek önerilerde bulunulmuştur.

2. KDS VE OLAP

2.1 Karar Destek Sistemleri (KDS)

Çalışma kapsamında yapılan literatür arařtırmaları; yöneticilerin karar verme sürecinde deęişik boyutlardan, hızlı ve özet bilgiye duydukları ihtiyacı ortaya çıkarmıştır. Bu ihtiyaçtan yola çıkarak çalışma, KDS'ler ve saęlık sektöründe uygulamalarına sık rastlanmayan OLAP uygulamaları üzerinde gerçekleştirilmiştir.

Yöneticilerin zamanında ve doęru karar verebilme istekleri, karar verme ortamı ile doğrudan ilişkilidir. Karar verme sürecinde, yönetime destek vermek amacıyla hedeflenen bilginin üretilmesi ve sunulması için kullanıcı etkileşimli yazılım ve donanım araçlarının bütünleşik kümesinden oluşan etkileşimli bilgi sistemleri Karar Destek Sistemleri (DSS), olarak adlandırılırlar (Atak 2006). Bu şekilde tanımlanan KDS'ler kullanıcının veriyi daha iyi analiz edebilmesini, etkinliğini arttırmayı ve analizi yapılan verilerin daha iyi anlaşılmasını hedefler. KDS'ler yöneticinin karar vermesini kolaylaştırır, fakat onun yerine çıkarımlarda bulunamaz. Kısaca, üretilen yazılım, karar verme noktasında yöneticinin etkinliğini arttırıyorsa KDS olarak deęerlendirilebilir.

Karar Destek Sistemleri tüm kurum bazında orta düzey yöneticileri içeren geniş bir kullanıcı grubu tarafından kullanılabilceęi gibi, üst düzey tek bir yöneticinin bilgisayarında kurulu olan bütünleşik bir yazılım sistemi de olabilir. Nasıl bir sistem kurulacaęı daha çok ihtiyaçlar ve hedefler çerçevesinde belirlenir.

Hastane gibi sürekli veri akışı içerisinde bulunan kurumlarda, yöneticiler kurulan bilgi yönetim sistemleri sayesinde elde ettikleri ham verileri, veri ambarları ve KDS yardımıyla etkin bir şekilde filtreleme işleminden geçirip, kullanışlı ve yararlı bilgiler haline getirmeyi, bu sayede stratejik önem arz eden konularda doęru ve hızlı karar vererek verimlilięi arttırmayı, saęlık hizmetlerinin süreklilięini saęlamayı ve hasta memnuniyetini arttırmayı hedeflerler. Maliyet-Kazanım dengesi, performans deęerleri, hasta sayıları, paket ve vaka başı uygulamalarının incelenmesi, satın alma-malzeme-stok takibi gibi konular hastane için önemli karar noktalarıdır. Yöneticilerin bu

noktalarda başarılı bir KDS ile desteklenmesi, etkin ve doğru karar vermeyi kolaylaştırarak, verimliliği artırır (Laudon ve Laudon 2002).

Yöneticiler karar verme aşamasında, HBYS'den bazı soruların cevabını bulmak isterler. Örneğin;

- Kış dönemi içerisinde performansı son üç yılın aynı dönemine göre en fazla yükselen bölüm hangisidir?
- Hangi malzemelerin kullanımını geçen yıla göre % 25'ten fazla artmıştır?
- Paket uygulama çerçevesinde değerlendirilen vakalarda, ilaçların toplam maliyete oranının bir önceki dönemle karşılaştırılması,
- Ayaktan Vaka Başı hastalarının bölümlere göre ortalama maliyet-kazanım dengesi,
- Bölümlere göre laboratuvar, radyoloji işlemlerinin dağılımının dönemsel olarak değişimi,
- Yeni alınan tıbbi cihazdan sağlanan kazanımlar nelerdir?

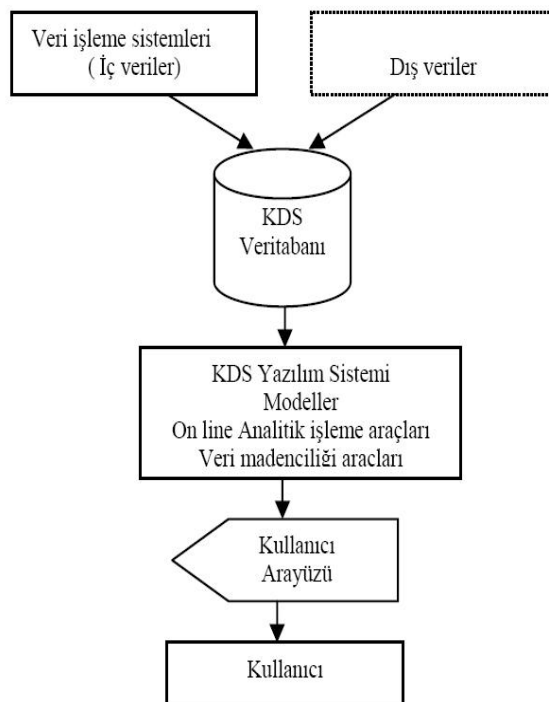
Bu soruların cevabının basit bir raporlama ekranından sürekli ve hızlı olarak elde edilmesi mümkün olmayabilir. Başarılı bir KDS, yöneticinin bu soruların cevabını bulmasına yardım eder.

KDS'ler, kullanıcılara ihtiyaç duydukları verileri kendi görüş açılarına göre düzenleyerek sunar. Bu sayede kullanıcıların istedikleri sonuca ulaşmalarını, basit konularda analiz yapmalarını veya karmaşık modellere dair ileriye yönelik tahmin yürütebilmelerini sağlar (WEB_1 2007).

Bir KDS, verilerin tutulduğu veritabanı, bu verilerin analiz işlemine uygun hale getirileceği yazılım sistemi ve kullanıcıların analiz yapabilmesini sağlayacak arayüzden oluşur.

Hastanede oluşturulacak bir karar destek sisteminin veri kaynağı elbetteki hastane bilgi yönetim sistemi olacaktır. Çalışanlar tarafından HBYS üzerinden elde edilen veriler, karar destek sisteminde kullanılmadan önce dördüncü bölümde anlatılan veri temizleme dönüştürme işleminden geçirilip, özetleme ve veri taşıma işlemiyle bir veri ambarında toplanmalıdır. Temizlenmiş, amaca yönelik sınıflandırılmış, özetlenmiş veri ambarındaki HBYS verileri analiz sistemleri tarafından ele alınır. Çalışmada OLAP

analiz yöntemi için Microsoft Analysis Services kullanılmıştır. Analiz sistemi, veri ambarından elde edilen verileri çok boyutlu, kullanıcının rahatlıkla yönetebileceği küpler haline getirir. OLAP küpleri halindeki verilerin kullanıcıyla buluşması, geliştirilen bir yazılımla sağlanmıştır. Bundan sonrası kullanıcıların isteklerine bağlıdır. Kullanıcı verilere istediği boyuttan bakarak normalde elde edilmesi zor, karmaşık hesaplamalara rahatlıkla ulaşabilir, analizi yaparak karar destek noktasında kendisine yardımcı olacak bilgiler elde edebilir. Şekil 2.1’de Karar Destek Sistemleri’nin temel bileşenleri gösterilmiştir.



Şekil 2.1 Karar Destek Sistemleri'nin temel bileşenleri (Çetinyokuş ve Gökçen 2002)

2.2 Online Analytical Processing(OLAP)

Herhangi bir organizasyona KDS kurmak, organizasyona ait yeterli veri toplanmasına, veri yapısının iyi tasarlanmış olmasına ve bu yapıdan yola çıkılarak veri ambarlarının ve veri marketlerinin oluşturulmasına bağlıdır. Toplanan bu verilerden oluşturulan veri ambarları üzerinden verileri çok boyutlu, esnek yapıda gösterebilen yazılıma OLAP diyoruz.

OLAP, aynı zamanda yöneticilerin ayrıntılı analiz yapabilmesi için gerekli olan stratejik bilgiye hızlı ve sürekli şekilde ulaşmasını sağlar.

2.2.1 OLAP özellikleri

OLAP, sorgulama ve raporlama yazılımlarından üç önemli özelliği ile ayrılmaktadır:

- Verilerin çok boyutlu gösterimi,
- Karışık hesaplamalar,
- Zaman yönelimli süreç kabiliyeti.

Verilerin çok boyutlu gösterimi:

İş modelleri, özellikleri gereği verilerin çok boyutlu gösterimine ihtiyaç duyarlar. Hastane için örnekleme gerekirse; zaman, malzeme, çalışanlar, hastalar, servisler gibi birçok boyut tanımlanabilir. Birçok seviye içeren zaman boyutunda malzeme kullanımlarının aylık, 3 aylık ve yıllık süreçlerde nasıl değiştiği gözlenebilir. Anabilim Dalı (ABD), doktor gibi seviyelerde performans değerleri takip edilebilir. Hastaların sosyal güvenlik kurumlarına, doğum yerlerine, cinsiyetlerine vb. gibi parametrelerine göre çıkarımlar elde edilebilir.

OLAP uygulamaları veritabanındaki verilere esnek ve kolay ulaşım sağlayarak, küp yapısında dilimler halinde görünüm kazandırır. Uygulamalar yöneticilere herhangi bir boyutta veya seviyede, eşit kolaylık ve işlevsellikle çaprazlama analizler yapabilme olanağı sağlamalıdır. Örnek olarak, herhangi bir birim veya klinikteki, herhangi bir malzeme tipinin ya da alt kategorisinin, 15 günlük veya altı aylık süreçte kullanım değerleri kolaylıkla elde edilebilmelidir. OLAP uygulamaları bu tip isteklere cevap verebilecek veri yapısı ve görünümünde olmaları gerekmektedir. Bu sayede, yöneticiler SQL kodları yazmak, kompleks tablo yapılarını ve ayrıntılı tablo ilişkilerini bilmek zorunda kalmazlar (WEB_3 2007).

OLAP'ta verilerin çok boyutlu gösterimi veri küpleri olarak adlandırılır. Tipik olarak bir küpün üç boyuta sahip olduğu düşünülse bile, iş modelleri içeren data küpleri bir çok boyut içerebilir. Bunun için Excel örneğini düşünebiliriz. Satır, sütun ve sayfaları boyut olarak kabul etmenin yanı sıra, aynı bilgisayarda açılan farklı Excel belgelerini ve farklı bilgisayarlarda açılmış farklı Excel belgelerini boyut olarak kabul edebiliriz.

Karışık hesaplamalar:

Bir çok OLAP uygulaması, küpün bir hiyerarşisi boyunca veya bir boyut bazında basit data hesaplamaları yapabilirken, bazıları da bundan daha kompleks özellik içeren toplamın yüzdesini alabilme, gerçekleşmemiş bir sonraki değeri tahmin edebilme gibi karar destek aşamasında büyük yararlar sağlayabilecek hesaplamalar yapabilir. Benzer şekilde matematiksel denklemleri ve kompleks algoritmaları kullanan, değişken ortalama ve yüzde olarak artış hızı hesaplamaları da elde edilebilir.

Zaman yönelimli süreç kabiliyeti:

Zaman neredeyse tüm OLAP uygulamaları için evrensel bir boyuttur. Zaman gözetilmeksizin herhangi bir iş modellemesinin yapılabilmesi çok zor olabilir. Zaman boyutu, iş sürecinin performansını değerlendirmeye ve karşılaştırmaya yardımcı olabilir (Bain vd 2000). Örneğin, Kulak Burun Boğaz bölümünün bu ay gösterdiği performans geçen ay gösterdiği performans ile karşılaştırabilir veya hastanenin son 6 aylık süreçte elde ettiği döner sermaye geçen yıl aynı dönemdeki ile değerlendirilebilir.

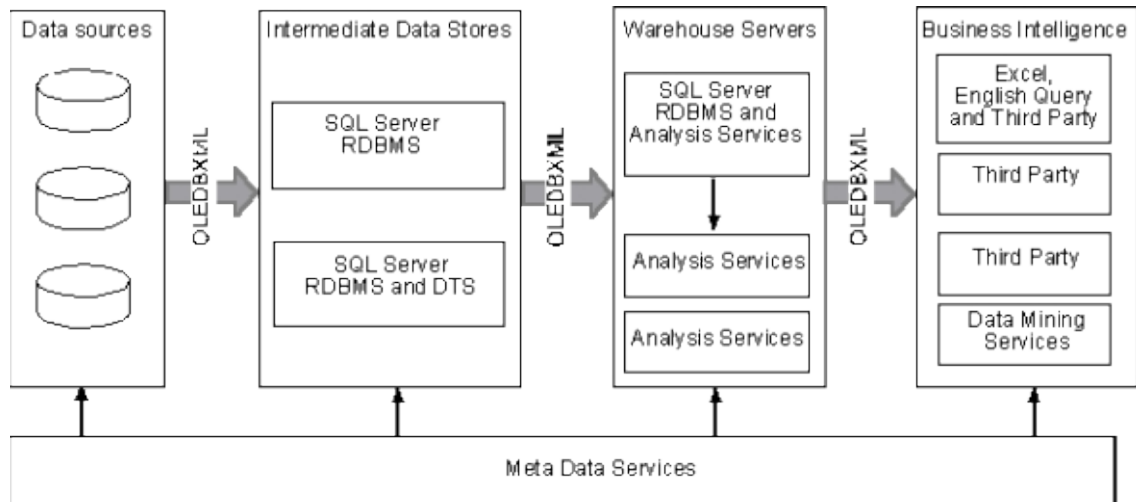
İlişkisel bir veritabanının karar destek faaliyetlerinde kullanılması performans açısından tavsiye edilmez. Geleneksel veritabanları yani Online Transaction Processing (OLTP) sistemler kayıt ekleme, güncelleme ve silme işlemlerinde iyi performans gösterirken, veri analizi konusunda OLAP sistemler uzmanlaşmıştır. OLAP ve OLTP arasındaki fark; OLTP sistemleri verileri toplayıp yönetirken, OLAP sistemleri toplanan verileri kullanarak yeni bilgilere ulaşılmasını sağlar.

KDS'lerde kullanılacak veri, değişik kaynaklardan veri temizleme, denormalizasyon gibi işlemlerden geçirilerek tek bir havuzda toplanabilir. Oluşturulan bu havuza veri ambarı (data warehouse) denir. Veri ambarları, sorgulama ve raporlama için tasarlanmış olup, en küçük birimler halinde ve özetlenmiş veriyi içerirler.

Verilerin değişik kaynaklardan çekilip veri ambarına aktarılması işlemine ise Extract Transform Load (ETL) denir. ETL; verinin farklı veri kaynaklarından alınıp ayrıştırılması (Extraction), iş analizleri yapabilmek için temizlenip, belirli formata dönüştürülmesi (Transformation) ve veri ambarlarına yüklenmesi (Loading) şeklinde tanımlanabilir. Analizi yapılacak veriler farklı veri kaynaklarında farklı formatlarda saklanıyor olabilir. Farklı kaynaklardaki verilerin ortak bir veri ambarında

toplanabilmesi için aynı formatta düzenlenmesi, denormalize edilmesi, veri tiplerinin ve yapılarının güncellenmesi gerekebilir. Bu işlemlerin tamamlanmasından sonra, farklı kaynaklardaki bu veriler analizin yapılacağı veri ambarına taşınarak, yüklenir. Çalışmada bu işlemleri gerçekleştirmek için dördüncü bölümde anlatılacak olan DTS kullanılmıştır.

Veri ambarlama ve OLAP birbirini tamamlayıcıdır. Büyük boyutlu kaynaklardan ve farklı bakış açılarında anlık veri talep etme noktasında bir araya gelirler. Genel olarak OLAP, standart veritabanında yapılamayacak derecede karmaşık olan sorguları gerçekleştirebilmek amacıyla veri ambarlarındaki ham verileri yüklenmekte, biçimlendirmekte, indekslemekte ve kullanıcının erişimine sunmaktadır. Şekil 2.2’ de SQL Server’ın veri ambarlama işleyişi gösterilmiştir.



Şekil 2.2 SQL Server Data Warehousing (WEB_2 2007)

OLAP, kullanıcılarını karmaşık ve dağınık veri yapısından kurtarıp nitelikli analizler yapabilmesini, dolayısıyla daha iyi çıkarımlara ulaşmasını sağlar. Ek olarak, kullanıcılar teknik bilgi ya da teknik eleman desteğine ihtiyaç duymaksızın verilerin çok boyutlu ve zaman süreçli hesaplanan değerlerine kolaylıkla ulaşarak kendi analizlerini oluşturabilirler.

OLAP sayesinde yöneticiler karmaşık problemlerin çözümünü bulabilir. Örneğin, başhekim, hasta sayısının düşüşünü belirli bir bölümdeki belirli bir tıbbi cihaz malzemesinin temin edilememe sorununa indirgeyerek, ilgili malzemenin son 6 aylık alış satış fiyatı değerlendirmesi ile sorunu inceleyebilir. Bununla birlikte OLAP

yöneticilere geleceğe yönelik kararlarda da destek sağlar. Örneğin alınacak ikinci bir tıbbi cihazın ücretli hasta miktarındaki artışa etkisi ne olabilir sorusuna cevap aranabilir.

2.2.2 OLAP'ın avantajları

- OLAP çok boyutlu esnek veri görünümü ile kullanıcıların veriyi daha iyi şekillendirebilmesini, analiz yapabilmesini ve çıkarımlarda bulunabilmesini sağlar.
- Kullanıcı dostudur. Birkaç mouse hareketi ile verilere tamamen farklı bir boyuttan bakarak farklı hesaplamaları değerlendirmek mümkündür.
- OLAP veri ambarından çıkartılan özet verileri indeksli ve çok boyutlu olarak tuttuğu için sorgu performansı çok yüksektir.
- OLAP verileri ilişkisel veritabanına bağlı olarak ya da olmayarak sadece özet veriyi ya da detayı ile birlikte tutabilir.
- OLAP basit faturalama raporlarından, karmaşık maliyet – tutar analizi hesaplamalarına kadar ilişkisel veritabanında sorgusu zor ve uzun olan hesaplamaları kolaylıkla yapabilir.

OLAP'ta veri, boyutları ile birlikte küp olarak adlandırılır. Bir OLAP veri küpü üzerinde şu işlemler yapılabilir:

- Dice: Boyutları çevirmek olarak ifade edilebilir. Örneğin, Paket Maliyet Analizi bilgisine zaman ve hasta türü boyutlarından bakarken küp görünümü değiştirilip cinsiyet ve yaş boyutundan bakılabilir.
- Slice: Dilimleme olarak ifade edilir. Tüm veriyi değil de sadece belli bir dilimi, örneğin hasta türünde sadece bağkurlu hastalar ele alınabilir.
- Drill down: Bir alt seviyeye geçiş. Örneğin, yer boyutunda Denizli ilindeki hasta sayısı gözlenirken ilçelerdeki dağılıma inilebilir (Immon 2000).
- Drill Up: Bir üst seviyeye geçiş. Örneğin, zaman boyutunda aylık fatura tutarları incelenirken 3 aylık, 1 yıllık tutarlara geçiş yapılabilir (Karaca 2006).

OLAP ve küplerle ilgili daha ayrıntılı bilgi, karar destek modülü oluşturulurken OLAP yazılımı olarak kullanılan Microsoft Analysis Services'in anlatıldığı beşinci bölümde verilecektir.

3. HASTANE BİLGİ YÖNETİM SİSTEMİ (HBYS)

Bütün işletmelerde olduğu gibi sağlık kurumlarında da, hem sunulan hizmet ile ilgili fonksiyonlar, hem de bizzat organizasyonla ilgili yönetim, planlama, pazarlama, finans, muhasebe gibi fonksiyonlarının yerine getirilmesi esnasında oluşan verilerin kaydı, kontrolü, arşivlenmesi, güvenliğinin sağlanması ve yeniden kullanılması otomasyonun temelini teşkil eder (WEB_5 2007).

Sağlık hizmetinin en iyi şekilde verilmesi, hizmetin sürekliliği, işlevsel akışkanlığın sağlanması, maliyetlerinin tespiti, hizmetlerin doğru ve hızlı yerine getirilmesi, hizmet karşılıklarının tahsili ve takibi, hizmet kalitesinin arttırılması, bilgilere kısa zamanda doğru şekliyle ulaşılması, istatistiksel çalışmaların kolayca yapılması ve bilimsel veriler elde edilmesi, gerekli verilerin karar mekanizmalarına sunulması ve kullanılması gereksinimleri, sağlık kuruluşlarında yönetim bilişim teknolojilerinin uygulanmasını ve dolayısıyla kurumlarda otomasyonuna geçilmesini zorunlu hale getirmiştir. Sağlık bilgi yönetim sistemleri, hastane otomasyonu, hastane bilgisayar otomasyonu, hastane otomasyon sistemleri, tıbbi enformasyon sistemleri vb tanımlar, temelde sağlık kuruluşlarında üretilen hizmetlerin olabildiğince bilgisayar desteği ile gerçekleştirilmesini ifade eden eş anlamlı kavramlardır. Kısaca, hastanenin işletim kuralları içinde yönetilmesinde, hastaların tıbbi ve finansal takibinin yapılmasında, kurumla ilgili günün ve geleceğin kararlarının alınmasında fayda sağlayacak bir veritabanının en az hata ile optimum zamanda işlemlerin yapılmasını sağlayacak bir elektronik entegrasyonla kurulması işlemidir (WEB_5 2007).

Hasta kayıtlarının tutulması, hizmetlerin faturalanması, gelir-gider takibinin yapılması, kaçakların önlenmesi, hizmet kayıplarının engellenmesi, hastane kaynaklarının en verimli şekilde kullanılması, stok takibinin yapılması ve hastanenin uluslararası ve resmi ortak veri bütünlüğüne ulaşmasını sağlayacak kodlama sistemlerini içermesi HBYS'nin gereklerindedir (Anonim 2004, 2005).

3.1 HBYS'nin Yararları

Dört maddede incelenebilir:

- İdari, mali ve sağlık bilgilerinin takibini ve paylaşımını sağlamak,
- İstenilen her türlü istatistiki ve karar destek bilgisine en kısa sürede ulaşmak,
- Kağıt dolaşımını ortadan kaldırmak,
- Kaynakları optimum düzeyde kullanmak.

3.2 Genel HBYS Modülleri

- Hasta Kayıt-Arşiv Modülü
- Poliklinik Yönetim Modülü
- Eczane Modülü
- Malzeme, Stok Takip ve Demirbaş Modülü
- Laboratuvar Modülü
- Hasta Yatış-Çıkış Modülü
- Fatura ve Muhasebe Modülü
- Performans Modülü
- Sağlık Kurulu Modülü
- Radyoloji Modülü
- PACS (Picture Archiving and Communications System - Görüntü Arşivleme ve İletişim Sistemi)
- KARAR DESTEK MODÜLÜ
- Ameliyathane Modülü
- İnsan Kaynakları Modülü
- Klinik Yönetim Modülü
- Satın Alma Modülü
- Cihaz Yönetim Modülü (Anonim 2004, 2005)

HBYS'nin uygulamaya geçirilmesi ve sonrasında işletiminin devamı hastane içerisinde birçok dinamiği etkilediğinden sistem tasarımcıları ve uygulayıcıları için çeşitli zorluklar içermektedir: Bunlar;

- İş akışlarının belirlenmesindeki zorluklar,
- Kullanıcıların bilgisayar kullanım bilgisinin ve alışkanlığının yetersizliği,
- Veri giriş standardizasyonunun sağlanmasındaki zorluklar,
- Uluslararası ve resmi kodlama sistemlerinin kullanımının benimsetilmesindeki zorluklar olarak özetlenebilir.

Bu dört maddede özetlenen zorluklar, veri girişlerinde farklılığa ve veri kirliliğine neden olabilir. Dolayısıyla karar destek modülünden verim elde edilebilmesi, sistem tasarımcıları ve uygulayıcılarının başarısıyla doğrudan ilişkilidir diyebiliriz.

4. KARAR DESTEK MODÜLÜ VERİ AMBARININ OLUŞTURULMASI

Çalışma kapsamında; literatür araştırması tamamlandıktan, yöntemler ve araçlar belirlendikten sonra, bilgi akışına ihtiyaç duydukları konuları belirlemek amacıyla hastane yöneticileri ile görüşüldü. Yapılan kapsamlı analiz sonucunda karar destek modülünün alt konu başlıkları Pamukkale Üniversitesi Hastaneleri HBYS nin içerdiği veriler göz önünde bulundurularak belirlendi.

Sonraki aşamada KDS'nin oluşturulması işlemine geçildi. Verimli bir KDS kurulması için HBYS'ler de toplanan verilerin, etkin bir temizliğe tabi tutulması ve gerekli parametrelere göre seçim sürecinden geçirilmesi ve elde edilen özet verilerin veri ambarına aktarılması gerekir.

4.1 Veri Temizleme İşlemi

HBYS'ler gibi çok fazla verinin yoğun bir şekilde işlendiği ortamlarda veri giriş elemanlarının dalgınlığı, daha önce kullanılan sistemlerle yeni sistem arasındaki veri giriş farklılıkları ve özellikle de yazılımların kullanıcıyı formatlı veri girişine zorlamaması veri kirliliğine yol açmaktadır. Karar Destek Sistemlerinde kullanılan parametrelerin aynı bilgiyi değişik formatlarda tekrarlamaması amacıyla veri temizleme ve dönüştürme işleminin gerçekleştirilmesi gerekmektedir.

Veri temizliği altı noktada önem kazanmaktadır. Bunlar;

- Veride eksik bilgilerin bulunması,
- Standart formatın dışında saklanmış verilerin bulunması,
- Çelişen bilgilerin bulunması,
- Yanlış ya da geçerliliğini yitirmiş verilerin bulunması,
- Aynı bilginin değişik şekillerde tekrarlanması,
- Bilgiler arasında bütünlüğü oluşturacak ilişkinin bulunmaması (WEB_4 2006, WEB_6 2007).

Veri kalitesinin artırılması amacıyla yukarıda bahsedilen maddelerden yola çıkılarak, Pamukkale Üniversitesi Hastaneleri HBYS Microsoft SQL Server 2000 üzerinde bulunan veritabanı tabloları üzerinde veri temizleme işlemleri yapıldı. Bu işlem için kullanılan SQL ifadeleri EK-2’de verilmiştir.

4.2 Karar Destek Modülü Raporları

Veri Temizleme işlemi tamamlandıktan sonra verilerin istenilen raporlara göre özetlenmiş şekilde tutulacağı bir veri ambarı oluşturuldu. Veri ambarı, HBYS veritabanının zarar görmemesi, mevcut çalışmaların aksamaması ve aynı zamanda karar destek modülü son kullanıcısının sorgulama hızının artması için farklı sunucuda konumlandırıldı. Bu işlem için; 2 adet Intel Dual Core Xeon 5130 2.0 GHz hızında işlemci, 4 GB kapasiteli bellek, 2 adet 146 GB veri saklama kapasitesi, konfigürasyonuna sahip bir sunucu temin edildi.

Sunucu üzerinde HospitalTEMP adıyla bir veritabanı oluşturularak HBYS’deki veriler biçimlendirilip, özetlenerek HospitalTEMP veritabanına Microsoft SQL Server 2000 Microsoft Data Transformation Services (DTS) yardımıyla taşındı.

Microsoft SQL Server 2000 Data Transformation Services:

Microsoft SQL Server’ın veri dönüşüm ve taşıma işlemi gerçekleştiren servisidir. Çalışmada SQL Server ın ilişkisel veritabanından elde edilen verileri veri ambarı haline getirerek analize sunulmasını sağlar. Genellikle SQL Server veritabanları üzerinde çalıştırılrsa da Oracle gibi değişik veri tabanı sistemlerinden, text dosyalarından, Excel belgelerinden veri alışverişini de destekler (Bain vd 2000).

DTS çalışması gerçekleştirilirken öncelikle karar destek noktalarında analiz yapılabilmesi için HBYS’deki ilgili veriler belirlendi. Bu verilerin ilişkisel tablolardan çıkarımı için SQL ifadeleri ve kullanıcı tanımlı fonksiyonlar oluşturuldu.

Verilerin taşınacağı HospitalTEMP veritabanında tutulması için gerekli tablo (fact table) yapıları da tasarlandı. Bu tabloların analiz sisteminde kullanılacak OLAP küpleri için dimension ve measure bilgilerini içerecek alanları belirlendi.

Aşağıda yapılan bu işlemlerin bilgisi verilmektedir.

4.2.2 Paket hastaların bölüme göre dağılım raporu

Bütçe Uygulama Talimatnamesine (BUT) göre bazı işlemler paket olarak değerlendirilip bu fiyat üzerinden faturalandırma işlemi yapılır. Hastalara yapılan işlemlerin toplamı ve geri ödeme kurumları tarafından ödenecek olan paket bedelleri karşılaştırılarak paket hastaların maliyet-kazanım analizi yapılır. Hastane hizmetlerinin etkin ve verimli devam edebilmesi için paket hasta bedelleriyle gerçek maliyetlerin yönetimce belirlenmiş bir dengede tutulması gerekir. Karar noktasında paket işlemlerin hangi hasta profillerinde ne kadara mal olduğu bilgisi değişkenlik gösterdiğinden hastaların tanısı, yaşı ve yatış süresi bilgileri önem arz etmektedir.

Oluşturulan Dimension Bilgileri:

SevkTarih
 Hasta Türü
 Yatış Süresi
 Anlık Yaş
 Cinsiyet
 Bölüm

Oluşturulan Measure Bilgileri:

Geliş Sayısı	Count (Refno)
Yatış Süresi	sum(CikisTarihi -YatisTarihi)
Maliyet	sum(I.YtlHiztutar)
Tutar	sum(RP.YtlTBorc)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

RefNo, SevkTarih, Cinsiyet, Yas, HastaTuru, Bolum,
 YatisSuresi, Tutar, Maliyet

```
DROP TABLE [PaketNormalKarsilastirma]

CREATE TABLE [PaketNormalKarsilastirma] (

[Refno] decimal (18,0) NOT NULL,

[Bolum] char (12) NULL,
```

```
[Sevktarih] datetime NULL,
[Cinsiyet] char (5) NULL,
[HastaTuru] char (10) NULL,
[AnlikYas] int NULL,
[YatisSuresi] int NULL,
[Tutar] decimal (18,2) NULL,
[Maliyet] decimal (18,2) NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
PaketTutariBul
CREATE function dbo.KDS_PaketTutariBul(@refno as int = 0)
returns decimal(18,2) as
BEGIN Declare @Tutar as decimal(18,2)
Set @Tutar = 0.00
  (Select @Tutar = sum(I.YtlHiztutar) From TH_ISLEM I WITH
(NOLOCK)
Where I.Refno = @Refno and Left(I.Hizkod,1) = 'P' and Fattipi
= '10' and Anahtar<>'-1')
if @Tutar is null
Select @Tutar = YTLTBorc from Th_Refkart where Refno = @refno
return @Tutar END
Select R.Refno, R.Polkod as Bolum, R.Sevktarih,
  (select Cinsiyet from TH_Dosya where Dosyano = R.Dosyano) as
Cinsiyet,
  (select HastaTuru from TH_Dosya where Dosyano = R.Dosyano) as
HastaTuru,
  (select Cast(datediff(year,Dogtarih,R.SevkTarih) as int) from
TH_Dosya WITH (NOLOCK) where Dosyano = R.Dosyano) as AnlikYas,
Cast(datediff(day,R.YatTarih,R.ÇıkTarih)+1 as int) as
YatisSuresi, dbo.KDS_PaketTutariBul (RP.Refno) as Tutar, (Select
sum(I.YtlHiztutar) From TH_ISLEM I WITH (NOLOCK) where I.Refno =
R.Refno and Fattipi = '10' and Anahtar<>'-1') as Maliyet from
TH_REFKART RP WITH (NOLOCK) inner join TH_REFKART R WITH
(NOLOCK) on RP.Refno = R.Paketrefno where RP.Paket = 1 and
RP.Aktif<>'2'
```

4.2.3 Pakete dönüştürülen referansların hizmet türüne göre dağılım raporu

Pakete dönüştürülen Referanslar üzerinden hizmet türüne göre maliyet hesaplamalarının yapılabilmesi, bu sayede paketlerin hastane gelir gider dengesi içinde durumunun ortaya konabilmesi için yardımcı olabilecek bir rapordur. Bir önceki raporda belirtilen genel paket hasta maliyeti bilgisine, hizmet türü bazlı detaylı bir bakış sağlamaktadır. Örneğin Paket uygulamalarda maliyet üzerinde belirleyici rol oynayan malzeme kullanımı bu raporla yakından izlenerek, malzeme kullanımının arttığı veya yüksek seyrettiği bölümlerde nedenler araştırılarak çözüm yolları aranmalı, gerekirse paket uygulama anlaşmalarının yeniden gözden geçirilmesi konusunda anlaşmalı kurumlar ile iletişime geçilmelidir.

Oluşturulan Dimension Bilgileri:

Hizmet Türü

Bölüm

Hizmet Adı

İşlem Tarihi

Oluşturulan Measure Bilgileri:

Tutar Sum(Tutar)

Toplam Miktar Sum(Miktar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Key0, Refno, IslemTarihi, Birim, Hizad, Hizmetturu, Miktar, Tutar

```
DROP TABLE [PaketNormalHizmetturuDetay]
```

```
CREATE TABLE [PaketNormalHizmetturuDetay] (
```

```
[Key0] decimal (18,0) NOT NULL,
```

```
[Refno] decimal (18,0) NULL,
```

```
[IslemTarihi] datetime NULL,
```

```
[Birim] char (12) NULL,
```

```
[Hizad] char (75) NULL,
```

```
[Hizmetturu] char (10) NULL,
[Miktar] decimal (18,0) NULL,
[Tutar] decimal (18,2) NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select I.Key0, I.Refno, I.Tarih as IslemTarihi,
I.istekyapanbirim as Birim, I.Hizad, H.Hizmetturu, I.Hizmiktar
as Miktar, I.YtlHizTutar as Tutar from Th_Islem I WITH (NOLOCK)
inner join Th_HizmetKart H on I.Hizkod = H.Hizkod where I.Refno
in (Select R.Refno from Th_Refkart R WITH (NOLOCK) where
R.PaketRefno is not null and R.Aktif<>'2') and I.Anahtar<>'-
1'and tarih> = '01.01.2005' and refno>600000
```

4.2.4 Paket ameliyatların bölüme göre dağılım raporu

Paket ameliyatların bölümüne, işlem tarihine ve hastanın sosyal durumuna göre değerlendirilmesi paket ameliyat profilinin çıkarılmasını kolaylaştırmaktadır. Uygulamadan kimlerin ne kadar faydalandığı, bölümlerin ameliyat dağılımları, ameliyatlara göre hastaların ortalama yatış süreleri gibi bilgilerinin çıkarımı yapılabilir.

Oluşturulan Dimension Bilgileri:

```
İşlem Tarihi
Hasta Türü
Anlık Yaş
Cinsiyet
Bölüm
```

Oluşturulan Measure Bilgileri:

```
Ameliyat Sayısı Count (Key0)
```

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

```
RefNo, SiraNo, IslemTarihi, HizmetAdi, Birim, HastaTuru,
Cinsiyet, AnlikYas
```

```
DROP TABLE [PaketAmeliyatBolum]
```

```
CREATE TABLE [PaketAmeliyatBolum] (
```

```
[RefNo] decimal (18,0) NULL,
[Sirano] decimal (18,0) NOT NULL,
[IslemTarihi] datetime NULL,
[HizmetAdi] char (75) NULL,
[Birim] char (12) NULL,
[HastaTuru] char (10) NULL,
[Cinsiyet] char (5) NULL,
[AnlikYas] int NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select RefNo, Key0 as Sirano, Tarih as IslemTarihi, Hizad as
HizmetAdi, istekYapanBirim as Birim, (Select D.HastaTuru From
Th_Dosya D inner join Th_Refkart R on D.Dosyano = R.DosyaNo
where R.Refno = I.Refno ) as HastaTuru, (Select Cinsiyet From
Th_Dosya D inner join Th_Refkart R on D.Dosyano = R.DosyaNo
where R.Refno = I.Refno ) as Cinsiyet, (Select
datediff(year,Dogtarih, R.SevkTarih) From Th_Dosya D inner join
Th_Refkart R on D.Dosyano = R.DosyaNo where R.Refno = I.Refno )
as AnlikYas from Th_Islem I WITH (NOLOCK) where hizkod like
'P6%' and Anahtar<>'-1' order by I.Refno
```

4.2.5 Hastaların ayaktan/yatan durumuna ve bölümüne göre dağılım raporu

Hastaların durumlarına ve hizmet aldıkları bölümlere göre verilen hizmet tutarlarının makbuz ve fatura toplamı olarak çıkarılmasını sağlar. Verilen hizmet tutarı, alınan karşılık değerlerinin takip edilmesini kolaylaştıran geniş çaplı bir rapordur. Bu raporla bölümlerin hasta sayısı ve tutar bazında performansları ölçülebilir, kesintilerin nedeni araştırılarak performanslara yansıtılabilir.

Oluşturulan Dimension Bilgileri:

Geliş Tarihi

Hasta Türü

Bölüm

Durum

Yatış Süresi

Oluşturulan Measure Bilgileri:

Hasta Sayısı	Count (Refno)
Hizmet Tutarı	Sum(YtltBorc)
Alinan	Sum(YtltAl)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

RefNo, SevkTarihi, Poliklinik, HastaTuru, Durum, YatisSuresi, Borc, Alinan

```
DROP TABLE [HastaDurumBolumGelirDagilim]
CREATE TABLE [HastaDurumBolumGelirDagilim] (
    [SevkTarih] datetime NULL,
    [Poliklinik] char (12) NULL,
    [Durum] char (10) NULL,
    [Refno] decimal (18,0) NOT NULL,
    [YatisSuresi] int NULL,
    [HastaTuru] char (10) NULL,
    [Borc] decimal (18,2) NULL,
    [Alinan] decimal (18,2) NULL)
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
YatisSuresiBul
Create Function dbo.KDS_YatisSuresiBul(@refno as int = 0)
returns integer as
BEGIN declare @YatisSuresi as integer
set @YatisSuresi = (Select datediff(day,R.YatTarih,
R.ÇıkTarih)+1 From TH_REFKART R where R.Refno = @refno)
if @YatisSuresi is null set @YatisSuresi = 0
return @YatisSuresi END
Select SevkTarih, Polkod As Poliklinik, Durum, Refno, Cast
(dbo.KDS_YatisSuresiBul (R.Refno) as int) as YatisSuresi,(Select
HastaTuru From Th_Dosya D where D.DosyaNo = R.DosyaNo) as
HastaTuru,YTltBorc as Borc, YtltAl as Alinan from Th_RefKart R
```

WITH (NOLOCK) where R.PaketRefno is null and R.Aktif<>'2' and YTLTBorc<>0 and SevkTarih> = '01.01.2004'

4.2.6 Faturaların kuruma göre dağılım raporu

Verilen hizmetlerin faturalanması ve geri ödemelerinin zamanında alınması hastane içindeki en önemli iş akışıdır. Bu akışın kesintisiz ve düzgün şekilde devam edebilmesi hizmetlerin sürekliliği için gereklidir. Hastaların Sosyal Güvencelerine ve Kurumlarına göre faturalama ve geri ödeme değerlerinin sürekli takibi bu raporla kolayca yapılabilir.

Oluşturulan Dimension Bilgileri:

Sevk Tarihi

Fatura Tarihi

Hasta Türü

Kurum Adı

Oluşturulan Measure Bilgileri:

Hasta Sayısı Count (Refno)

Fatura Tutarı Sum(YtlFatTutar)

Düşüm Tutarı Sum(YtlDüşTutar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

RefNo, SevkTarih, FaturaTarih, HastaTuru, KurumAdi, Tutar, Dusum

DROP TABLE [KurumFaturaGeriOdemeDagilim]

CREATE TABLE [KurumFaturaGeriOdemeDagilim] (

[RefNo] decimal (18,0) NOT NULL,

[SevkTarih] datetime NULL,

[FaturaTarih] datetime NULL,

[KurumAdi] char (100) NULL,

[HastaTuru] char (10) NULL,

[Tutar] decimal (18,2) NULL,

[Dusum] decimal (18,2) NULL)

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
FaturaTutariBulAvbli

CREATE function dbo.KDS_FaturaTutariBulAvbli(@refno as int =
0, @avakabasi as int = 0)

returns decimal(18,2) as

BEGIN Declare @Tutar as decimal(18,2)

Set @Tutar = 0.00

if      @avakabasi = 1

Select @Tutar = (Select Sum(YtlFatTutar) From FTR_FaturaGenel
F where F.Refno = @Refno and Fattipi = 10 )

else if @avakabasi = 0

Select @Tutar = (Select Sum(AVBOTutar) From FTR_FaturaGenel F
where F.Refno = @Refno and Fattipi = 10 )

return @Tutar END

Select RefNo, Sevktarih, FatTarih as FaturaTarih, KurAd as
KurumAdi,

(Select HastaTuru From Th_Dosya D where D.DosyaNo =
R.DosyaNo) as HastaTuru, dbo.KDS_FaturaTutariBulAvbli(R.Refno,
R. AvakaBasi) as Tutar, (Select sum(YtlDüşTutar) From
FTR_FaturaDusum FD where FD.Refno = R.Refno ) as Dusum from
Th_RefKart R WITH (NOLOCK) where R.PaketRefno is null and
R.aktif<>'2' and YTLTBorc<>0 and Sevktarih> = '01.01.2004' and
R.FatDurum>1
```

4.2.7 Yatan hastaların tanı ve ABD'na göre dağılım raporu

Yatan hastalar gerek maliyet, gerekse hizmet sıklığı açısından ayaktan hastalara göre artı değerler içermektedir. Sağlık hizmetinin yoğun olarak verildiği yatan hastaların, ABD ve tanıya göre gruplu takip edilmesi klinik karar destek ve maliyet açısından önem taşımaktadır.

Oluşturulan Dimension Bilgileri:

Geliş Tarihi

Servis

Yatış Süresi

Birincil Tanı

Oluşturulan Measure Bilgileri:

Muayene Sayısı Count(Islemsirano)

Fatura Tutarı Sum(YtltBorc)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Islemsirano, TaniAdi, Servis, SevkTarih, Tutar, YatisSuresi

DROP TABLE [YatanHastaBolumTaniDeger]

CREATE TABLE [YatanHastaBolumTaniDeger] (

[Islemsirano] numeric NOT NULL,

[TaniAdi] varchar(255) NOT NULL,

[Servis] char(12) NULL,

[SevkTarih] datetime NULL,

[Tutar] decimal(18, 2) NULL,

[YatisSuresi] int NULL)

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

Select Rf.islemsirano, Rf.icdAd as TaniAdi, Rf.Polkod as
Servis,

(Select SevkTarih from Th_Refkart R where R.Refno = Rf.Refno)
as SevkTarih,

(Select YTLTBorc from Th_Refkart R where R.Refno = Rf.Refno)
as Tutar,

(dbo.KDS_YatisSuresiBul(Rf.Refno)) as YatisSuresi From
Th_RefTani RF

WITH (NOLOCK) where Rf.islemsirano>0 and Rf.ontani = 0 and
Rf.birinciltani = 1 and Rf.refno in (Select refno from
Th_Refkart WITH (NOLOCK) where Aktif<>'2' and Yattarih is not
null)


```
Rf.islemsirano>0 and Rf.ontani = 0 and Rf.birinciltani = 1 and  
Rf.Refno in (Select refno from Th_Refkart WITH (NOLOCK) where  
Yattarih is null and Aktif<>'2')
```

4.2.9 Tanı hasta dağılım raporu

Hastaneye müracaat eden hastalar tanılarına göre değerlendirilmek istenirse bu rapordan faydalanılabilir. Belirli zaman aralıklarında gelen hastalarda, x tanısının y değeri üzeri ve altı yaşlarda, z cinsiyetlerinde sık görülmesi ve buradan yola çıkılarak uygulanan tedavilerin, muayene ve epikriz notlarının incelenmesi sağlık hizmetinin kalitesinin artırılması açısından artı değerler getirebilir.

Oluşturulan Dimension Bilgileri:

Geliş Tarihi

Hasta Türü

Cinsiyet

Yaş

Birincil Tanı

Oluşturulan Measure Bilgileri:

Muayene Sayısı Count(islemsirano)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

SevkTarih, islemsirano, TaniAdi, HastaTuru, Cinsiyet,
AnlikYas

```
DROP TABLE [TaniHastaProfil]  
CREATE TABLE [TaniHastaProfil] (  
[islemsirano] decimal (18,0) NOT NULL,  
[TaniAdi] varchar (255) NOT NULL,  
[SevkTarih] datetime NULL,  
[HastaTuru] char (10) NULL,  
[Cinsiyet] char (5) NULL,  
[AnlikYas] int NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select Rf.islemsirano, Rf.icdAd as TaniAdi, (Select SevkTarih
from Th_Refkart R where R.Refno = Rf.Refno) as SevkTarih,
(Select D.Hastaturu From Th_Dosya D inner join Th_Refkart R on
D.Dosyano = R.Dosyano where R.Refno = Rf.Refno) as
HastaTuru, (Select Cinsiyet From Th_Dosya D inner join Th_Refkart
R on D.Dosyano = R.Dosyano where R.Refno = Rf.Refno) as
Cinsiyet, (Select datediff(year,Dogtarih,R.SevkTarih) From
Th_Dosya D inner join Th_Refkart R on D.Dosyano = R.Dosyano
where R.Refno = Rf.Refno ) as AnlikYas From Th_RefTani Rf WITH
(NOLOCK) where Rf.islemsirano>0 and Rf.ontani = 0 and
Rf.birinciltani = 1 and Rf.Refno in (Select refno from
Th_Refkart WITH (NOLOCK) where Aktif<>'2')
```

4.2.10 Hizmet türüne ve bölüme göre dağılım raporu

Ülkemizde sağlık sektöründe en çok uygulanan fiyat tarifesi olan Bütçe Uygulama Talimatına(BUT) göre hizmetler belirli başlıklar altında gruplandırılmıştır. Hangi bölümün hangi hizmet türünden ne kadar işlem isteği gerçekleştirdiği, tarih bazlı performansı, gereksiz ve aynı işlemlerin istenme sıklığı bu raporla belirlenebilir. Bölümler bazında hizmet türlerinin belirlenmesi ihtiyaçların daha iyi analiz edilebilmesini sağlar. Örneğin, Ortopedi ve Beyin Cerrahisi bölümlerinde hasta sayısının artması, buna bağlı olarak yeni bir poliklinik açılması kararı için, takip eden süreçte radyoloji ve grafi hizmet türündeki isteklerin artması ve bu isteklerin ne kadarının hangi zaman diliminde karşılanabildiği bilgisi yeni bir tıbbi cihaz alımı kararı için önemli destek noktaları oluşturur.

Oluşturulan Dimension Bilgileri:

Hizmet Türü

Bölüm

Hizmet Adı

İşlem Tarihi

Oluşturulan Measure Bilgileri:

Tutar Sum(YtlHizTutar)

Hasta Sayısı Count (Refno)

Miktar Sum(HizMiktar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Refno, Islemtarihi, HizmetAdi, Birim, HizmetTuru, Miktar, Tutar

```
DROP TABLE [HizmetTuruBolumProsedurDeger]
CREATE TABLE [HizmetTuruBolumProsedurDeger] (
[Refno] decimal (18,0) NULL,
[islemTarihi] datetime NULL,
[HizmetAdi] varchar (75) NOT NULL,
[Birim] char (12) NULL,
[Miktar] decimal (18,0) NULL,
[Tutar] decimal (18,2) NULL,
[HizmetTuru] char (10) NOT NULL,
[Anahtar] nvarchar (15) NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select Refno, Tarih as islemTarihi, H.Hizad as HizmetAdi,
İstekYapanBirim as Birim,Hizmiktar as Miktar, YtlHizTutar as
Tutar, HizmetTuru, (Select Ad From Th_ProsedurStepTanim where
Pkod = HizmetTuru and Kod = Anahtar ) as Anahtar From Th_Islem I
WITH (NOLOCK) inner Join Th_HizmetKart H WITH (NOLOCK) on
I.HizKod = H.Hizkod where I.Refno not in (Select Refno From
Th_Refkart WITH (NOLOCK) where AVakabasi = 1 or Paket = 1 or
Aktif = '2' or YTLTBorc = 0 or (Fatno = '0' and YtlBakiye<>0) )
and tarih> = '01.01.2005' and refno>600000
```

4.2.11 Tedavisi tamamlanmış hastaların uzman hekim onayı alınmamış radyoloji, laboratuvar, patoloji işlemlerinin dağılım raporu

Bu rapor herhangi bir bölüme muayene olup doktor tarafından gerekli görülen tetkik ve tahlilleri istenen hastanın tedavisini tamamlamasına rağmen eksik kalan onaylanmamış işlemlerinin dökümünün alınmasından oluşur. Bu sayede hizmeti verilmiş olmasına rağmen onayı verilmediği için fatura edilemeyecek olan işlemlerin hastane içindeki dağılımı ve onay verilmeme sebepleri belirlenir.

Olusturulan Dimension Bilgileri:

Hizmet Türü

Hizmet Adı

İşlem Tarihi

Step Durum

Olusturulan Measure Bilgileri:

Tutar Sum(YtlHizTutar)

Miktar Sum(HizMiktar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Islemtarihi, HizmetAdi, HizmetTuru, Anahtar, Miktar, Tutar

DROP TABLE [HizmetTuruBolumProsedurDeger]

CREATE TABLE [HizmetTuruBolumProsedurDeger] (

[Refno] decimal (18,0) NULL,

[islemTarihi] datetime NULL,

[HizmetAdi] varchar (75) NOT NULL,

[Birim] char (12) NULL,

[Miktar] decimal (18,0) NULL,

[Tutar] decimal (18,2) NULL,

[HizmetTuru] char (10) NOT NULL,

[Anahtar] nvarchar (15) NULL)

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select Refno, Tarih as islemTarihi, H.Hizad as HizmetAdi,
İstekYapanBirim as Birim,Hizmiktar as Miktar, YtlHizTutar as
Tutar, HizmetTuru, (Select Ad From Th_ProsedurStepTanim where
Pkod = HizmetTuru and Kod = Anahtar ) as Anahtar From Th_Islem I
WITH (NOLOCK) inner Join Th_HizmetKart H WITH (NOLOCK) on
I.HizKod = H.Hizkod where I.Refno not in (Select Refno From
Th_Refkart WITH (NOLOCK) where AVakabasi = 1 or Paket = 1 or
```

```
Aktif = '2' or YTLTBorc = 0 or (Fatno = '0' and YtlBakiye<>0) )
and tarih> = '01.01.2005' and refno>600000
```

4.2.12 Hastaların sosyal güvencelerine göre fatura değerlerinin dağılım raporu

Hastaya verilen hizmetin tutarı vaka başı, paket, katkı payı gibi uygulamalardan dolayı fatura tutarı olarak değerlendirilemez. Hastaların sosyal güvencelerine göre faturalama değerlerinin çıkarılması ve bunların takibi gerektiğinde basit bir raporlama olarak kullanılabilir.

Oluşturulan Dimension Bilgileri:

Hasta Türü

Geliş Tarihi

Oluşturulan Measure Bilgileri:

Hasta Sayısı Count (Refno)

Fatura Tutarı Sum(YtlFatTutar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Refno, Sevktarih, HastaTuru, FaturaTutari

DROP TABLE [FaturaSosyalGuvence]

CREATE TABLE [FaturaSosyalGuvence] (

[Refno] decimal (18,0) NOT NULL,

[Sevktarih] datetime NULL,

[FaturaTutari] decimal (18,2) NULL,

[HastaTuru] char (10) NULL)

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select     Refno,        Sevktarih,        dbo.KDS_FaturaTutariBulAvbli
(R.Refno,R.AvakaBasi) as Tutar, (Select HastaTuru From Th_Dosya
D where D.DosyaNo = R.DosyaNo) as HastaTuru From Th_Refkart R
WITH (NOLOCK) where PaketRefno is null and Aktif<>'2' and
YTLTBorc<>0 and Sevktarih> = '01.01.2004' and FatDurum>1
```


4.2.13 Uzman doktor/hasta türü katkı payı dağılım raporu

Uzman doktorların genelde aylık olarak çıkarılan katkı payları değerlerinin takip edilmesini sağlayan rapordur. Hastaların kendi isteğiyle muayene, ameliyat gibi sağlık hizmetlerini öğretim üyesi Uzman Doktorlardan almak istemeleri neticesinde tahsil edilir. Katkı Paylarının belirli bir miktarı vergi ve işletme gideri olarak kesilirken, kalanı hakediş olarak ilgili Uzman Doktora ödenir. Katkı payları üniversite hastanelerinde önemli derecede nakit girdisi sağladığı için uzman doktor ve hastanın sosyal güvencesine göre takip edilmesi gerekir.

Bu rapor kaynak veriyi oluşturan Fact Table tek olmasına rağmen, incelenmesi daha kolay olması amacıyla, farklı dimension ve measure'lara sahip iki farklı küpten oluşturulmuştur:

a) Hasta türüne göre;

Oluşturulan Dimension Bilgileri:

Hasta Türü

Makbuz Tarihi

Oluşturulan Measure Bilgileri:

Hasta Sayısı Count (Refno)

Katkı Tutarı Sum (YtlMakToplam)

b) Uzman Doktora göre;

Oluşturulan Dimension Bilgileri:

Makbuz Tarihi

Doktor Adı

Hizmet Adı

Oluşturulan Measure Bilgileri:

Hasta Sayısı Count (Refno)

Katkı Tutarı Sum(YtlMakToplam)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

```

Refno, MakbuzTutar, HizmetAdi, MakbuzTarih, Doktor, HastaTuru
DROP TABLE [KatkiPayiDeger]
CREATE TABLE [KatkiPayiDeger] (
  [Refno] decimal (18,0) NULL,
  [MakbuzTutar] decimal (18,2) NULL,
  [HizmetAdi] nvarchar (255) NULL,
  [Doktor] varchar (46) NULL,
  [MakbuzTarih] datetime NULL,
  [HastaTuru] char (10) NULL )

```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```

Select      Refno,      YtlMakToplam      as      MakbuzTutar,
Substring(MakAdı,1,charindex('  (  ',MakAdı)) as HizmetAdi,
(select MakTarih From THS_TahsilatGenel T where T.Makno =
TD.Makno) as MakbuzTarih, (Select Rtrim(DrUnvan)+' '+DrAdSoy
From Th_Doktor D where D.DrKod = TD.ServisYapan) as Doktor,
(select D.HastaTuru From Th_Dosya D where D.Dosyano in (Select
R.Dosyano From Th_Refkart R where R.Refno = TD.Refno)) as
HastaTuru from THS_TahsilatDetay TD WITH (NOLOCK) where
ServisYapan like 'DR.%' and Refno>500000 and ServisYapan <>
'DR.102'

```

4.2.14 Hasta makbuz bilgileri dağılım raporu

Hastaneye makbuz karşılığı ödeme yapmış hastaların ödeme şekillerini, sosyal güvencelerini, durumlarını inceleyerek analiz yapmak yöneticileri ilginç sonuçlara ulaştırabilir. Örneğin Yeşil Kartlı hastaların ne kadar katkı payı yatırdığı, kredi kartı ödemelerinin nakit ödemelere göre son bir yılda ne kadar arttığı, ayaktan hastalarda mı yoksa yatan hastalarda mı ödemenin fazla olduğu gibi bilgiler değerlendirilebilir.

Oluşturulan Dimension Bilgileri:

Makbuz Tarihi

Hasta Türü

Kredi Kartı Nakit

Hasta Durum

Oluşturulan Measure Bilgileri:

Hasta Sayısı Count (Refno)

Makbuz Tutarı Sum(YtlMakToplam)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Refno, MakbuzTutar, NakitKredi, HastaDurum, MakbuzTarih,
HastaTuru

```
DROP TABLE [HastaMakbuzTutar]
```

```
CREATE TABLE [HastaMakbuzTutar] (
```

```
[Refno] decimal (18,0) NULL,
```

```
[MakbuzTutar] decimal (18,2) NULL,
```

```
[NakitKredi] char (1) NULL,
```

```
[HastaDurum] char (10) NULL,
```

```
[MakbuzTarih] datetime NULL,
```

```
[HastaTuru] char (10) NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select Refno, YtlMakToplam as MakbuzTutar,
```

```
(Select Tur From THS_TahsilatGenel T where T.Makno =  
TD.Makno) as NakitKredi,
```

```
(Select R.Durum From Th_Refkart R where R.Refno = TD.Refno)  
as HastaDurum,(Select MakTarih From THS_TahsilatGenel T where  
T.Makno = TD.Makno) as MakbuzTarih,(Select D.HastaTuru From  
Th_Dosya D where D.Dosyano in (Select R.Dosyano From Th_Refkart  
R where R.Refno = TD.Refno)) as HastaTuru from THS_TahsilatDetay  
TD WITH (NOLOCK) where Refno>500000
```

4.2.15 Vaka başı hastaların bölüme göre dağılım raporu

1 Temmuz 2006 tarihinde uygulamaya konulan Ayaktan Vaka Başı ödeme sistemi ile ayaktan hastaların hastaneye gerçek maliyeti ve kurumların ödediği bedeller arasında farklar oluşmuştur. Hastanenin Ayaktan Vaka Başı uygulamasından zarar etmemesi için takibinin sürekli ve bölüm bazında ele alınması gerekir. Hesaplama yapılırken BUT'ta EK 10C bölümüne girerek vaka başı değerlendirmenin dışında tutulan işlemlerin, fatura edilme sırasında vaka başı tutarına eklendiği göz önüne alınmalıdır.

Oluşturulan Dimension Bilgileri:

Poliklinik

Hasta Türü

Geliş Tarihi

Kurum Adı

Oluşturulan Measure Bilgileri:

Hasta Sayısı Count (Refno)

Maliyeti Sum (YtlFatTutar)

Tutarı Sum (AVBOTutar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Refno, SevkTarih, Poliklinik, HastaTuru, Tutar, Maliyet, KurumKodu

```
DROP TABLE [VakaBasiMaliyetTutarDeger]
```

```
CREATE TABLE [VakaBasiMaliyetTutarDeger] (
```

```
[Refno] decimal (18,0) NOT NULL,
```

```
[SevkTarih] datetime NULL,
```

```
[Poliklinik] char (12) NULL,
```

```
[HastaTuru] char (10) NULL,
```

```
[Tutar] decimal (18,2) NULL,
```

```
[Maliyet] decimal (18,2) NULL,
```

```
[KurumKodu] char (50) NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select Refno, Sevktarih, Polkod As Poliklinik, (Select
HastaTuru From Th_Dosya D where D.DosyaNo = R.DosyaNo) as
HastaTuru, (Select Avbotutar from FTR_FaturaGenel F where
F.Fatno = R.Fatno) as Tutar, (Select YTLFattutar from
FTR_FaturaGenel F where F.Fatno = R.Fatno) as Maliyet, (Select
KurAd from TH_Kurumlar K where K.Kurkod = R.Kurkod ) as
KurumKodu from Th_RefKart R WITH (NOLOCK) where R.Aktif<>'2' and
YTLTBorc<>0 and Sevktarih> = '01.01.2004' and avakabasi = '1'
```

4.2.16 Bölüme göre öğretim üyeleri performans uygulaması analizi

2547 sayılı Yükseköğretim Kanununun 58. maddesinin a fıkrasının 4. paragrafında, 17/9/2004 tarih ve 5234 sayılı Kanunun 2. maddesi ile yapılan değişikliklerden sonra Öğretim Üyelerine verilen performans hesaplaması doğrudan ve dolaylı olarak iki bölümden oluşur. Doğrudan performans, işlemlerin direkt öğretim üyesi tarafından yapılması neticesinde isim girişi yapılarak hesaplanır. Dolaylı performansta ise öğretim üyesi işlem yapılırken başında bulunmayabilir. Ancak işlemler daha sonra öğretim üyesinin onayından geçirilerek, doktor ismi girişine bakılmaksızın hesaplanır. Hesaplamalar sonucunda tahakkuk edilen tutar doğrudan işlemlerde performans esas tutarın, yönetimce belirlenen bir oranda, daha çok mesai dışındaki çalışma saatinin toplam çalışma saatine oranı, dolaylı işlemlerde ise bu oranın yarısından oluşur. Performans hesaplamaları kompleks ve uzun zaman alan bir işlemdir. OLAP çözümüyle bu karmaşıklık ve uzun süreç, kolay ve daha kısa zamana indirgenebilir.

a) Doğrudan Performans

Oluşturulan Dimension Bilgileri:

İstekYapanBirim

Hasta Türü

İşlem Tarihi

Doktor Adı

Oluşturulan Measure Bilgileri:

Hasta Sayısı

Count (Refno)

Tutar Sum(YtlHizTutar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Refno, Tarih, Birim, HastaTuru, Tutar, Doktor

```
DROP TABLE [PerformansDogrudan]

CREATE TABLE [PerformansDogrudan] (
  [Refno] decimal (18,0) NOT NULL,
  [Birim] char (12) NULL,
  [Tarih] datetime NULL,
  [Tutar] decimal (18,2) NULL,
  [PerformansTutar] decimal (18,2) NULL,
  [HastaTuru] char (10) NULL,
  [Doktor] varchar (46) NULL )
```

Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select P.Refno, I.istekyapanbirim as Birim, I.Tarih,
I.YtlHizTutar as Tutar, (SELECT CAST(REPLACE(H.perfyuzde, ',', '.')
as float)/100 * I.ytlhiztutar) as PerformansTutar, (Select
D.HastaTuru From Th_Dosya D inner join Th_Refkart R on D.DosyaNo
= R.DosyaNo where R.Refno = P.Refno) as HastaTuru, (Select
Rtrim(DrUnvan) +' '+DrAdSoy From Th_Doktor D where D.DrKod =
P.SeryapanDrkod) as Doktor From Th_Performans P inner join
Th_Islem I on P.Key0 = I.Key0 inner join Th_Hizmetkart H on
P.Hizkod = H.Hizkod and I.Anahtar<>'-1'
```

b) Dolaylı Performans

Oluşturulan Dimension Bilgileri:

İstekYapanBirim

Hasta Türü

İşlem Tarihi

Doktor Adı

Oluşturulan Measure Bilgileri:

Hasta Sayısı

Count (Refno)

Tutar Sum(YtlHizTutar)

Fact Table'ı oluşturmak için belirlenen alanlar ve gerekli SQL ifadesi:

Refno, Tarih, Birim, HastaTuru, Tutar, Doktor

```
DROP TABLE [PerformansDolayli]
```

```
CREATE TABLE [PerformansDolayli] (
```

```
[Refno] decimal (18,0) NULL,
```

```
[Birim] char (12) NULL,
```

```
[Tarih] datetime NULL,
```

```
[Tutar] decimal (18,2) NULL,
```

```
[Doktor] varchar (46) NULL,
```

```
[HastaTuru] char (10) NULL )
```

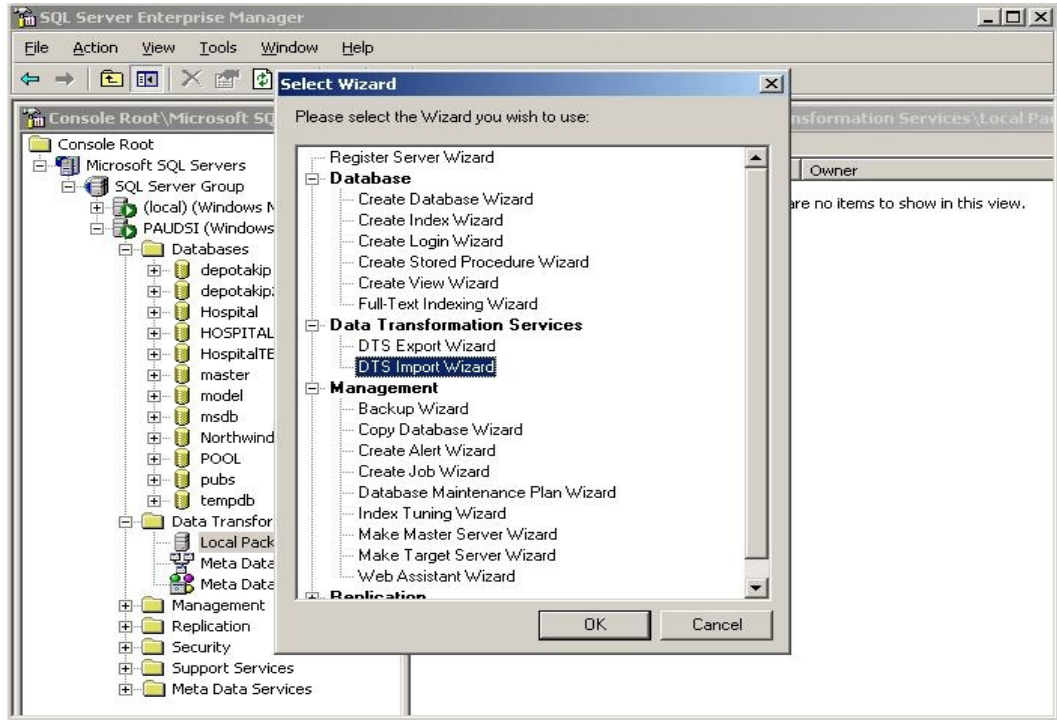
Fact Table'a atanan veriye ulaşmayı sağlayan SQL ifadesi:

```
Select I.Refno, I.istekyapanbirim as Birim, I.Tarih,
I.YtlHizTutar as Tutar, (Select R.trim(DrUnvan)+' '+DrAdSoy From
Th_Doktor D inner join Th_PerformansKonsultanHekim PKH on
PKH.Polkod = I.istekyapanbirim and PKH.DrKod = D.DrKod and
Year(I.Tarih) = PKH.Yil and Month(I.Tarih) = PKH.Ay and
DATEPART(week, I.Tarih) = PKH.Hafta) as Doktor, (Select
D.HastaTuru From Th_Dosya D inner join Th_Refkart R on D.DosyaNo
= R.DosyaNo where R.Refno = I.Refno) as HastaTuru From Th_Islem
I inner join Th_HizmetKart H on I.Hizkod = H.Hizkod and
perfparam = 1 and HizmetTuru not in ('YATAK','KONS') Where
I.Key0 not in (Select P.Key0 From Th_Performans P) and I.Tarih>
= '01.03.2006'
```

4.3 DTS Paketlerinin Oluşturulması

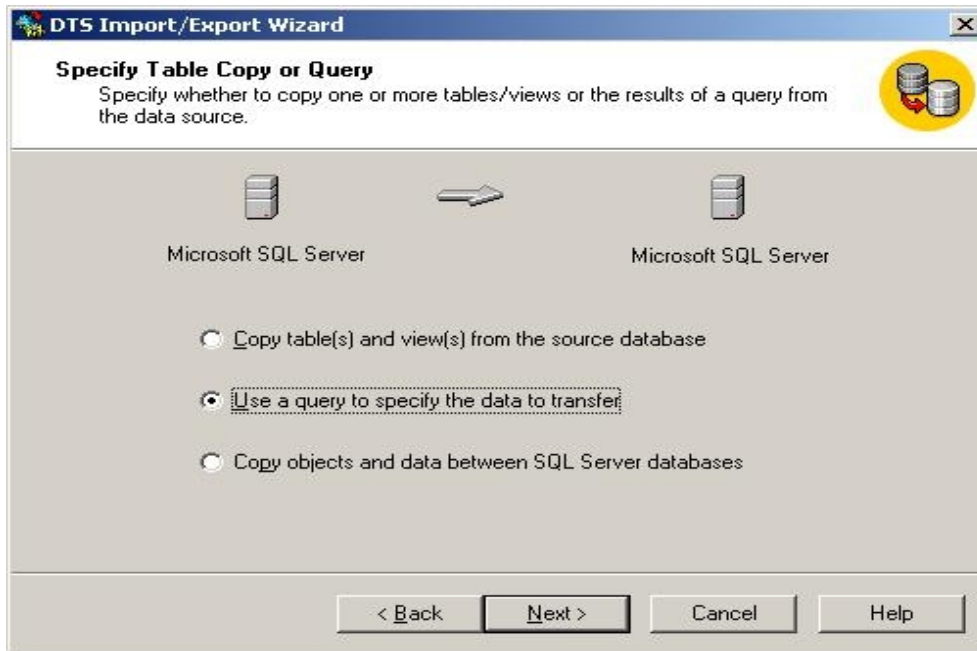
Küp yapılarının belirlenmesinin ardından verilerin HBYS server daki veritabanından, veri ambarının oluşturulduğu ve analizin yapılacağı server daki HospitalTEMP veritabanına taşınma işlemine geçilmiştir.

Bu işlem, SQL Server Enterprise Manager ekranında Data Transformation Services sekmesindeki Local Packages altından manuel tanımlanabileceği gibi, SQL Server Enterprise Manager'ın Tools menüsü altındaki Wizard yardımıyla da yararlanılabilir. Şekil 4.1'de SQL Server Enterprise Manager Wizard ekranı görülmektedir.



Şekil 4.1 SQL Server Enterprise Manager Wizard ekranı

Veri taşıma işlemi için öncelikle DTS Import Wizard ekranından kaynak HBYS veritabanı ve hedef HospitalTEMP veritabanı seçilir. Wizard bir sonraki ekranda kullanıcının veri taşıma yöntemini belirlemesini ister.

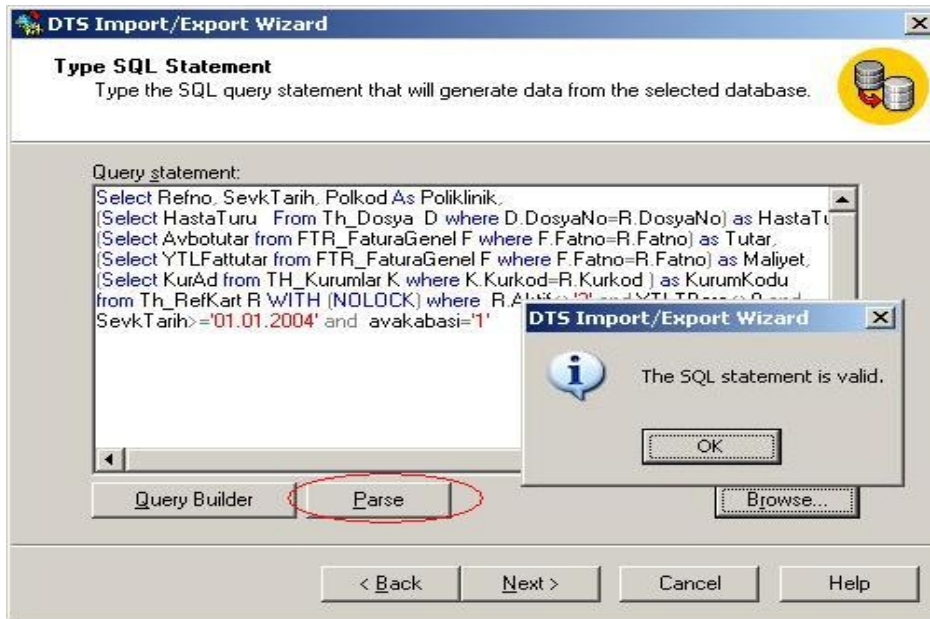


Şekil 4.2 SQL Server Enterprise Manager DTS veri taşıma yöntemi

Şekil 4.2’de de görüldüğü gibi Wizard, DTS veri taşıma yöntemini belirlemesi için kullanıcıya üç seçenek verir:

- Table ve view’ların kaynak veritabanından hedef veritabanına direk taşınması
- Transfer edilmek istenen verinin kaynak veritabanından sorgular yardımıyla çekilmesi
- Kaynak veritabanının hedef veritabanına istenen verilerin ve table, index, user, stored procedure gibi nesnelere seçilerek tümünden taşınması

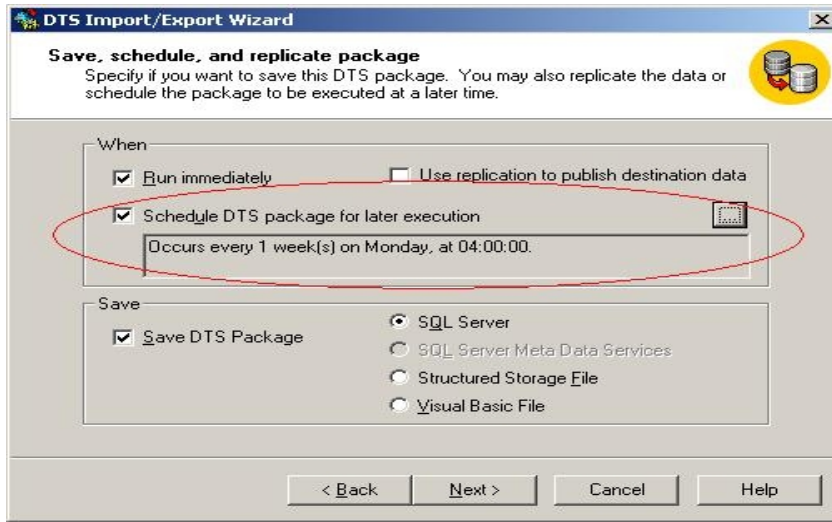
Veri ambarına taşınacak verilerin HBYS’deki bir çok tablodan değişik formatlarda çekilmesi gerektiğinden, çalışmada taşınacak verinin sorgu yardımıyla belirlenmesi yöntemi benimsenmiştir. Bölüm 4.2’de verilen, Fact Table’a atanan veriye ulaşmayı sağlayan SQL ifadesi, DTS Wizard’da belirlenen query ekranına girilir ve parse butonuna basılır. Şekil 4.3’de SQL Server Enterprise Manager DTS sorgu girme ekranı gösterilmiştir.



Şekil 4.3 SQL Server Enterprise Manager DTS sorgu girme ekranı

Hedef veritabanında istenen Fact Table’ın oluşturulması ve kaynak-hedef transformasyon bilgilerinin seçilmesi sonucunda, Wizard’ın son adımı olarak, oluşturulan DTS paketinin kaydedilmesi ve zamanlanması kullanıcının seçimine

sunulur. Şekil 4.4’de SQL Server Enterprise Manager DTS paketinin kaydedilmesi ve zamanlanması ekranı gösterilmiştir.



Şekil 4.4 DTS paketinin kaydedilmesi ve zamanlanması ekranı

4.4 DTS Paketlerinin Zamanlanması

Oluşturulacak veri ambarının sürekli güncel verilerle beslenmesi ve veri taşıma işleminin HBYS'nin yoğun olmadığı zamanlarda gerçekleştirilmesi için DTS Paketlerinin kaydı yapılmış ve çalışma zamanları hafta sonu geç saatlere ayarlanmıştır. Zamanlanmış görevlerin gerçekleştirilmesi için SQL Server Agent servisi kullanılmıştır.

SQL Server Agent: SQL Server Agent MS SQL Server 2000'in zamanlanmış görevleri yerine getirme servsidir. Backup, performans takibi, veritabanı bakımı, veri transferi gibi yönetimsel görevleri, belirlenen zamanda istenen şekilde otomatik olarak çalıştırır.

Şekil 4.5 DTS Wizard zamanlama ekranından da görülebileceği gibi kullanıcı oluşturulan DTS paketini, belli periyotlarda otomatik çalışmasını sağlayacak şekilde zamanlayabilir. Bu işlem SQL Server Agent'da kayıtlı bir job oluşturacaktır. Zamanlanan job manuel olarak durdurulana kadar belirlenen zaman ve saat dilimde çalışır. Ancak kullanıcı bitiş zamanı vererek oluşturacağı job'ın çalışmasını sınırlandırabilir.

Edit Recurring Job Schedule

Job name: New Package

Occurs: Daily Weekly Monthly

Weekly: Every 1 week(s) on: Mon Tue Wed Thur Fri Sat Sun

Daily frequency: Occurs once at: 00:04:00 Occurs every: 1 Hour(s) Starting at: 00:00:00 Ending at: 23:59:59

Duration: Start date: 02.12.2007 End date: 02.12.2007 No end date

OK Cancel Help

Şekil 4.5 DTS Wizard zamanlama ekranı

DTS paketinin isim bilgisi ve istenirse owner password ve user password'ü girilerek HBYS veritabanının bulunduğu sunucuya kaydı yapılır. Oluşturulan DTS paketlerinin ve zamanlama bilgilerinin görüntüleri Şekil 4.6'da verilmiştir.

Name	Description	Owner	Create Date
11HastaKimlikBilgileri		HASTANE\mtaser	07.06.2007 10:04:47
12PaketNormalKarsilastirma		HASTANE\mtaser	07.06.2007 10:05:53
13PaketNormalHizmetTuruDetay		HASTANE\mtaser	07.06.2007 10:06:11
14PaketAmeliyatBolum		HASTANE\mtaser	07.06.2007 10:06:55
15HastaDurumBolumGelirDagilim		HASTANE\mtaser	07.06.2007 10:07:40
16KurumFaturaGeriOdemeDa...		HASTANE\mtaser	07.06.2007 10:08:12
17YatanHastaBolumTaniDeger		HASTANE\mtaser	07.06.2007 10:10:34
18AyaktanHastaBolumTaniDe...		HASTANE\mtaser	07.06.2007 10:12:44
19TaniHastaProfil		HASTANE\mtaser	07.06.2007 10:13:19
20HizmetTuruBolumProsedurD...		HASTANE\mtaser	07.06.2007 10:14:00
21FaturaSosyalGuvence		HASTANE\mtaser	07.06.2007 10:14:41
22KatkıPayiDeger		HASTANE\mtaser	07.06.2007 10:15:14
23HastaMakbuzTutar		HASTANE\mtaser	07.06.2007 10:15:51
24VakaBasiMaliyetTutarDeger		HASTANE\mtaser	07.06.2007 10:16:24
25PerformansDogrudan		HASTANE\mtaser	07.06.2007 10:17:10
26PerformansDolayli		HASTANE\mtaser	07.06.2007 10:17:41

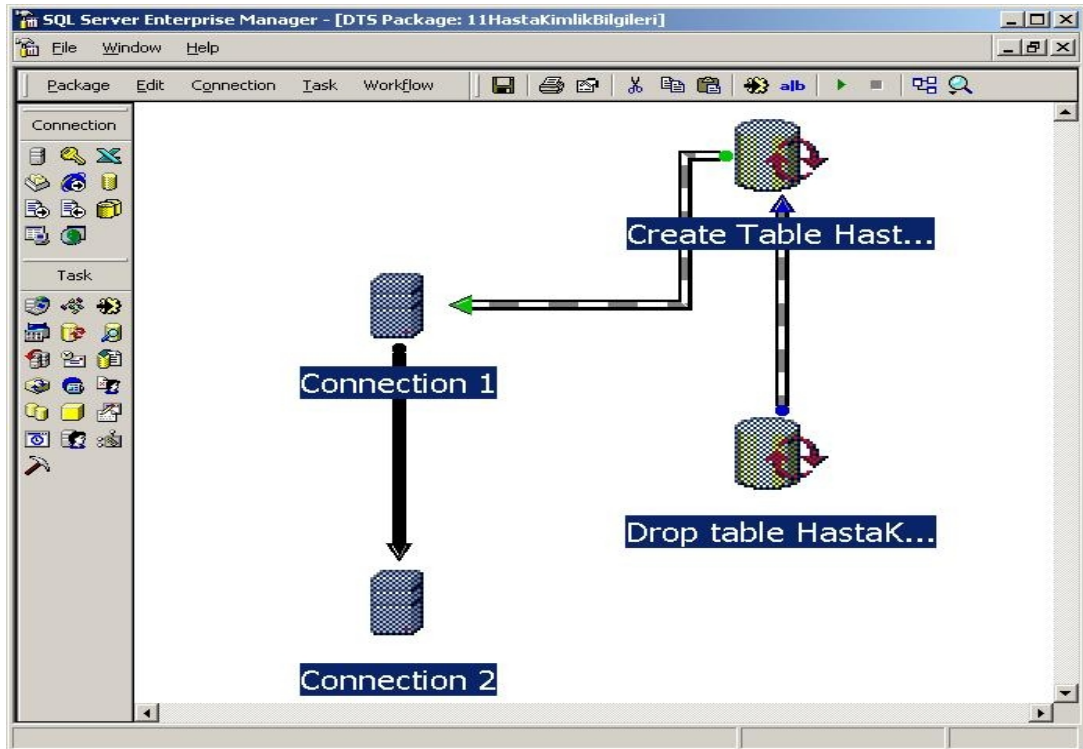
Şekil 4.6 DTS Packages

DTS paketlerinin zamanlanması sonucunda oluşan SQL Server Agent Jobs listesi Şekil 4.7'de gösterilmiştir.

Name	Last Run Status (Start Date)	Next Run Date
DTS_11HastaKimlikBilgileri	Succeeded (03.12.2007 02:00:00)	(10.12.2007 02:00:00)
DTS_12PaketNormalKarsilastirma	Succeeded (03.12.2007 02:10:00)	(10.12.2007 02:10:00)
DTS_13PaketNormalHizmetTuruDetay	Succeeded (03.12.2007 02:10:01)	(10.12.2007 02:10:00)
DTS_14PaketAmeliyatBolum	Succeeded (03.12.2007 02:20:00)	(10.12.2007 02:20:00)
DTS_15HastaDurumBolumGelirDagilim	Succeeded (03.12.2007 02:30:00)	(10.12.2007 02:30:00)
DTS_16KurumFaturaGeriodemeDagilim	Succeeded (03.12.2007 08:58:01)	(10.12.2007 02:40:00)
DTS_17YatanHastaBolumTaniDeger	Succeeded (03.12.2007 03:00:00)	(10.12.2007 03:00:00)
DTS_18AyaktanHastaBolumTaniDeger	Succeeded (03.12.2007 09:11:48)	(10.12.2007 03:10:00)
DTS_19TaniHastaProfil	Succeeded (03.12.2007 03:20:00)	(10.12.2007 03:20:00)
DTS_20HizmetTuruBolumProsedurDeger	Succeeded (03.12.2007 03:30:00)	(10.12.2007 03:30:00)
DTS_21FaturaSosyalGuvence	Succeeded (03.12.2007 04:00:00)	(10.12.2007 04:00:00)
DTS_22KatkiPayiDeger	Succeeded (03.12.2007 04:10:00)	(10.12.2007 04:10:00)
DTS_23HastaMakbuzTutar	Succeeded (03.12.2007 04:10:01)	(10.12.2007 04:10:00)
DTS_24VakaBasiMaliyetTutarDeger	Succeeded (03.12.2007 08:57:51)	(10.12.2007 04:20:00)
DTS_25PerformansDogrudan	Succeeded (03.12.2007 04:30:00)	(10.12.2007 04:30:00)
DTS_26PerformansDolayli	Succeeded (03.12.2007 04:40:00)	(10.12.2007 04:40:00)
HBYS_Hospital backupDIFF1	Succeeded (02.12.2007 17:00:00)	(03.12.2007 12:00:00)
HBYS_Hospital backupFULL1	Succeeded (02.12.2007 21:00:00)	(03.12.2007 21:00:00)
HBYS_Hospital backupLOG1	Succeeded (03.12.2007 08:45:00)	(03.12.2007 10:45:00)
HBYS_HospitalbackupFullUSB	Succeeded (03.12.2007 05:00:01)	(04.12.2007 05:00:00)
HBYS_HOSPITALSEC backup	Succeeded (02.12.2007 21:30:00)	(03.12.2007 21:30:00)
HBYS_HospitalTEST Backup	Succeeded (02.12.2007 22:00:00)	(03.12.2007 22:00:00)
HBYS_master backup	Succeeded (02.12.2007 22:30:00)	(03.12.2007 22:30:00)
HBYS_msdb backup	Succeeded (02.12.2007 22:30:01)	(03.12.2007 22:30:00)
Truncate Database - Hospital	Succeeded (30.11.2007 20:00:00)	(07.12.2007 20:00:00)

Şekil 4.7 SQL Server Agent Jobs listesi

Şekil 4.8’de ise oluşturulan bir DTS paketi gösterilmiştir.

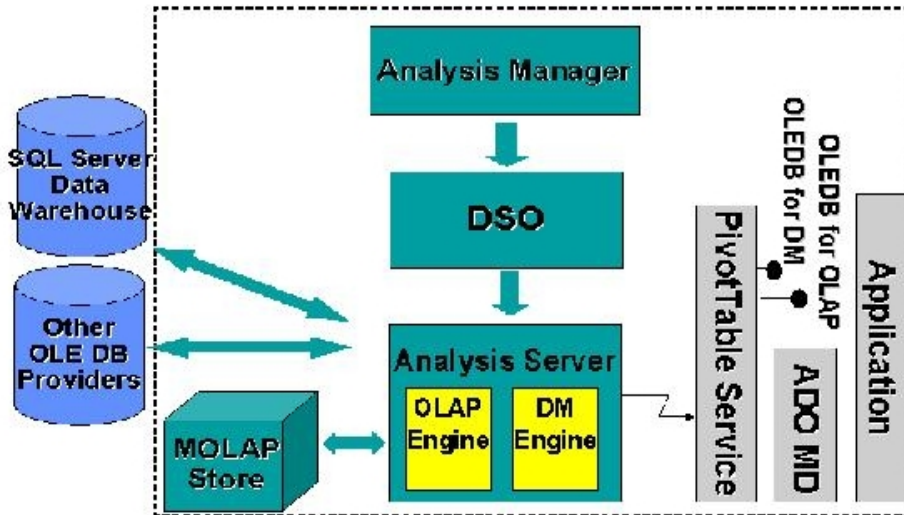


Şekil 4.8 Oluşturulan bir DTS paketinin görüntüsü

5. ANALİZ VERİTABANININ OLUŞTURULMASI

Sunucuda analiz işlemi için gerekli veri ambarı, HospitalTEMP veritabanı ve içerisinde küp verisinin kaynağı Fact Table'lar oluşturulduktan sonra OLAP küplerini içerecek analiz veritabanının oluşturulması işlemine geçilmiştir. Çalışmada analiz sistemi olarak Microsoft Analysis Services kullanılmıştır.

Microsoft SQL Server 2000 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 (Koorhan 2005). Şekil 5.1'de Analysis Services mimarisi gösterilmiştir.



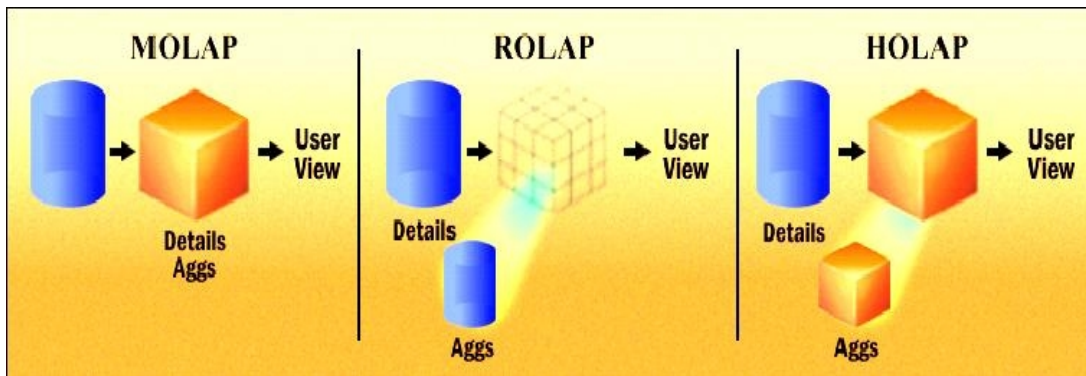
Şekil 5.1 Analysis Services Architecture (Microsoft Corporation 2000)

Analysis Services OLAP depolama şemalarının birçok tipini içerir. Bunlar;

Multidimensional OLAP (MOLAP): Asıl ve toplanan veriler küp yapısı içinde saklanır ve gerektiğinde oradan alınır. Bu yapı ilişkisel veritabanındaki veri kaynağı ile tamamen ayrılmıştır.

Relational OLAP (ROLAP): Asıl ve toplanan veriler veri kaynağı olan ilişkisel veritabanında saklanır. Bu depolama tipi, ilişkisel veritabanı avantajlarından ve enterprise data management araçlarından yararlanmayı sağlar.

Hybrid OLAP (HOLAP): Asıl veri ilişkisel veritabanında, toplanan veri ise multidimensional küp yapısında saklanır. HOLAP asıl verinin tekrar saklanmasına gerek kalmadan toplanan veri üzerinde MOLAP'ın sağladığı avantajlardan yararlanmayı sağlar (Tiedrich 2000). Şekil 5.2'de OLAP depolama şemaları gösterilmiştir.



Şekil 5.2 OLAP depolama şemaları (Microsoft Corporation 2000)

Çalışmada OLAP'ın hız ve esneklik seçeneklerinden tam olarak yararlanmak amacıyla MOLAP yöntemi kullanılmıştır.

5.1 Analysis Services Öğeleri

5.1.1 Analysis server

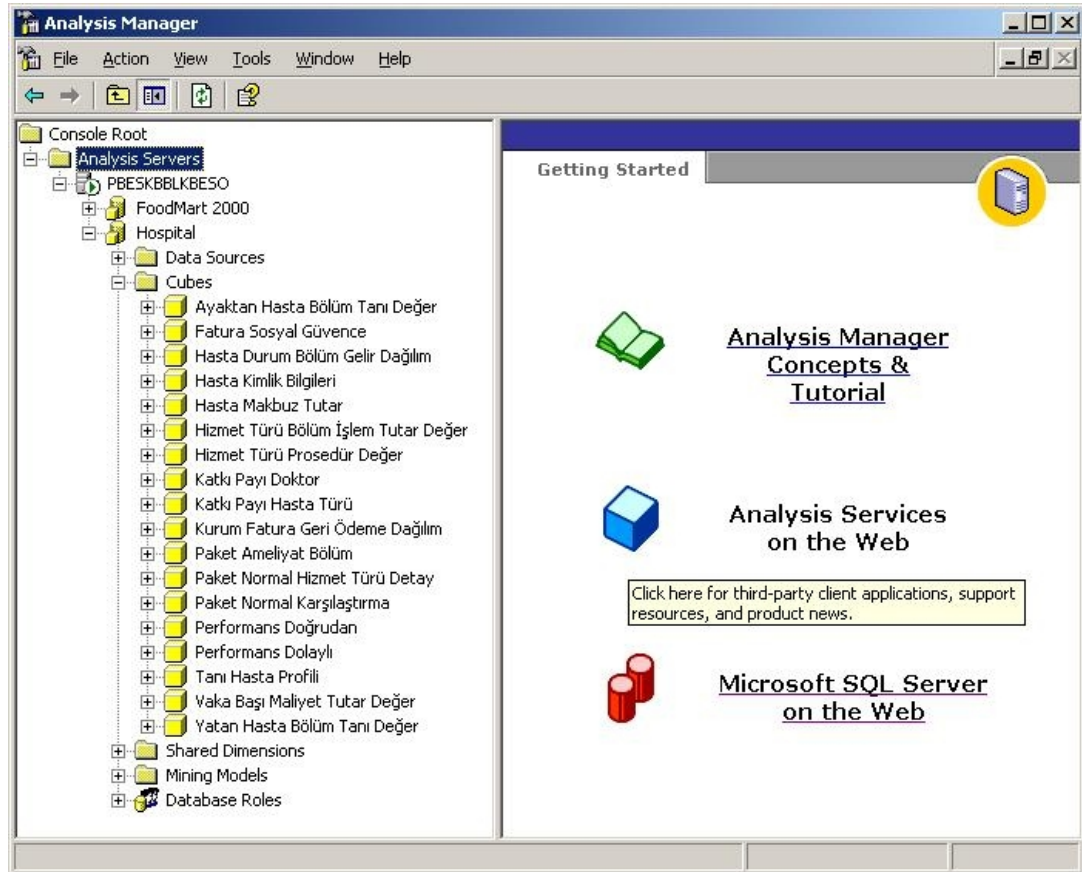
Microsoft SQL Server 2000 Analysis Services ürününün çekirdek bileşenidir. Nesne hiyerarşisindeki ilk nesnedir. Oluşturulan OLAP veritabanlarını, küpleri ve data mining modellerini yönetir.

5.1.2 Analysis manager

Analysis Server'ın ve verilerin grafiksel yönetim yazılımıdır. Bu araç Analysis Server üzerinden OLAP veritabanı ile bağlantı kurar. Analysis Server'lardan sonra oluşturulan nesnelere ise Analiz veritabanlarıdır (Bain vd 2000).

5.1.3 Analiz veritabanı

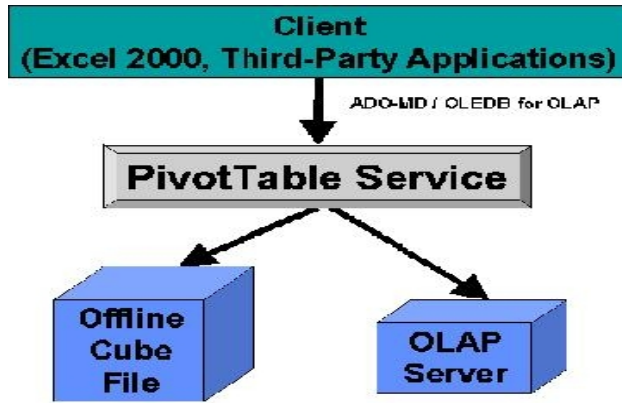
Data source'ları, küpleri, dimension'ları ve data mining modellerini içeren ve paylaşan veritabanıdır. İlişkisel HBYS veritabanından DTS yardımıyla HospitalTEMP veritabanına aktarılan bilgiler, Hospital Analiz veritabanında OLAP küplerine dönüştürülmektedir. Şekil 5.3'te çalışmadaki Analysis Manager öğeleri gösterilmiştir.



Şekil 5.3 Çalışmadaki Analysis Manager öğeleri

5.1.4 Pivot table service

İstemci tarafına veri almak için kullanılan Analysis Services'in istemci bileşenidir ve PTS olarak adlandırılır. PTS Microsoft Excel Pivot Tables'la aynı şey değildir. Ancak Microsoft Excel Pivot Tables PTS kullanabilir. PTS bir servis gibi çalışmaz ancak OLAP veritabanına yapılan sorgulara yanıt vermek üzere kullanılabilir (Koorhan 2005). Şekil 5.4'te Pivot Table Service gösterilmiştir.



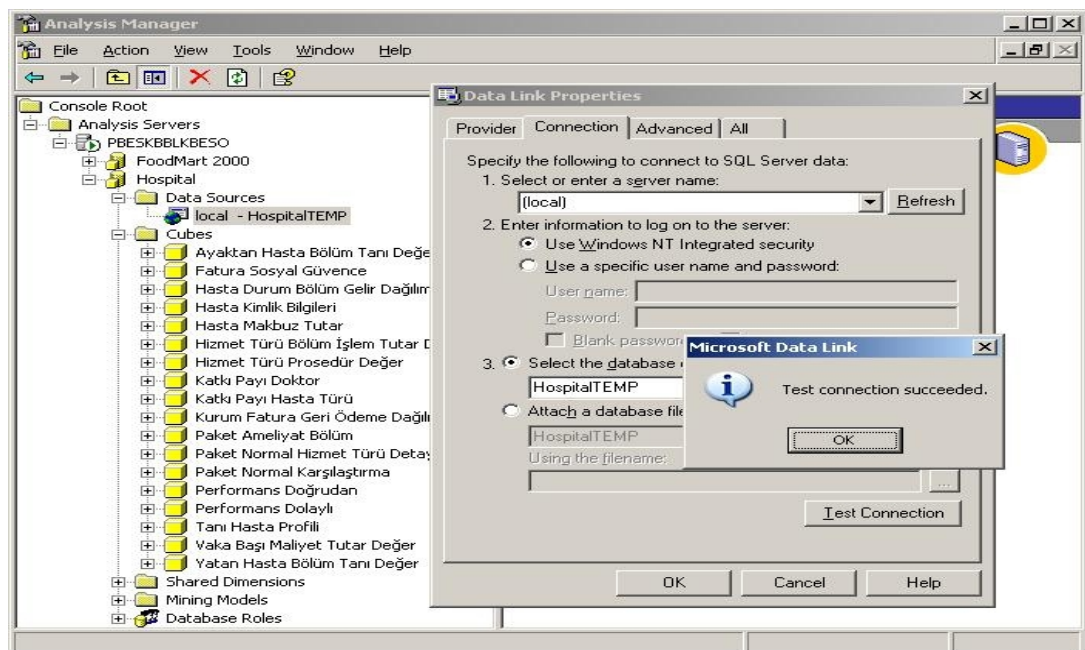
Şekil 5.4 Pivot Table Service (Microsoft Corporation 2000)

5.2 Analiz Veritabanı Öğeleri

5.2.1 Data source

Microsoft SQL Server 2000 Analysis Services ile kaynak veri ambarı arasındaki bağlantıdır. Analiz Server nesnelere verilere ulaşmak için Data Source'u kullanırlar (Bain vd 2000).

Şekil 5.5'te, analiz veritabanından veri kaynağı olarak kullanılan HospitalTEMP veritabanına bağlantı sağlamak için gerekli bağlantı parametreleri seçilerek bir data source oluşturulmaktadır.



Şekil 5.5 Data Source tanımı

OLAP tanımlamasında da bahsedildiği üzere OLAP veritabanlarındaki veriler çok boyutlu küpler şeklinde kullanıcıya sunulur.

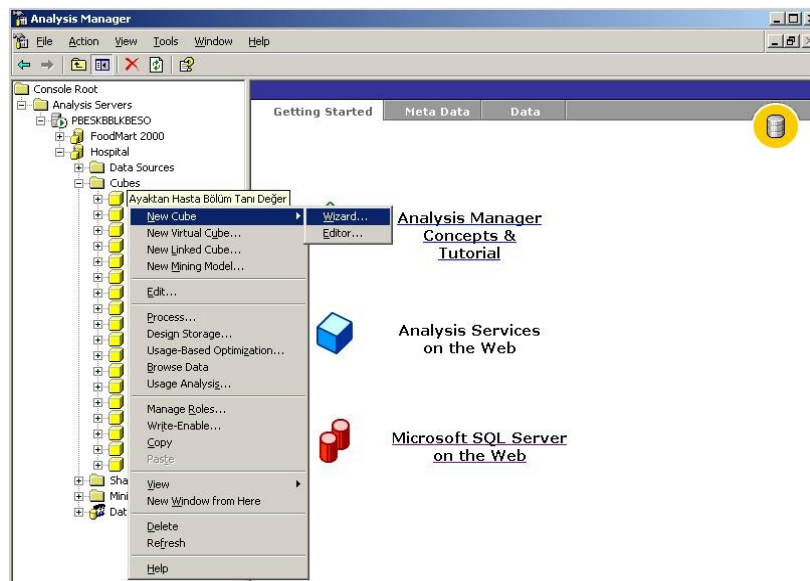
5.2.2 Cube (Küp)

Geleneksel doküman bazlı raporlardan farklı olarak OLAP küpleri veritabanında tutulan bilgiler içindeki pek çok etkileşimi dikkate alan, farklı yönlerden bakıldığında farklı konulara dikkat çeken, özel formatlı ve çok indeksli bilgi kaynaklarıdır.

Küpler, OLAP veritabanlarındaki mantıksal depolama yapılarıdır. Dimension ve measure'ları, esnek ve sezgisel modellerin içinde birleştirir ve kullanıcıların sorgu yaratmasını, verilerine ulaşmasını ve yönetmesini sağlar. Küpleri, dimension ve measure'ları içeren çok boyutlu yapı olarak da tanımlayabiliriz. Dimension'lar küpün yapısını oluştururken measure'lar son kullanıcıyı ilgilendiren nümerik verileri sağlar. Küplerdeki hücreler, dimensionlar arasındaki kesişim ile tanımlanır ve değeri measure'daki veri tarafından belirlenir.

5.3 OLAP Küpü Oluşturmak

Microsoft Analysis Manager yönetim konsolu kullanıcıları, OLAP küplerini kolaylıkla oluşturabilmek için New Cube Wizard'ı kullanabilirler. Bununla ilgili uygulama şekil 5.6 Analysis Manager New Cube Wizard ekranında gösterilmiştir.

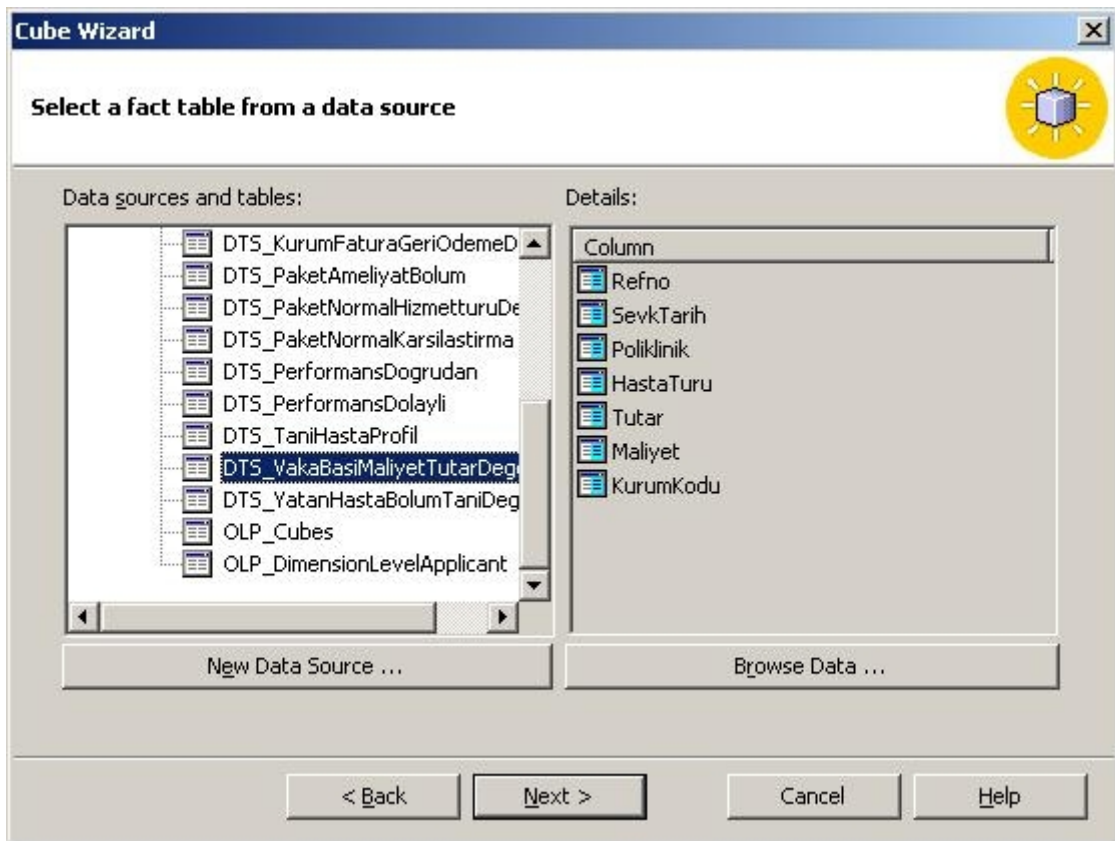


Şekil 5.6 Analysis Manager New Cube Wizard

Yeni Küp oluşturma sihirbazında ilk adım, küplerin veri kaynağı olarak kullanacağı Fact Table'ların seçimidir.

Fact Table'lar, analizi yapılmak istenen verileri içeren veri ambarındaki kaynak tablolarıdır. Bu tablolar büyük bir organizasyonun birkaç yıllık geçmişini içeren yüz milyonlarca kayda sahip olabilirler. Temelde Measures, Foreign Keys diye adlandırılan iki nesneye sahiptirler.

Fact Table her bir measure için bir sütun içerir. Ek olarak, ilişkili olduğu her bir Dimension Table'daki Primary Key'ler ile bağlantıyı sağlayan bir foreign key (Dimension Key) içerir. Şekil 5.7'de New Cube Wizard'da Fact Table seçimi gösterilmiştir.

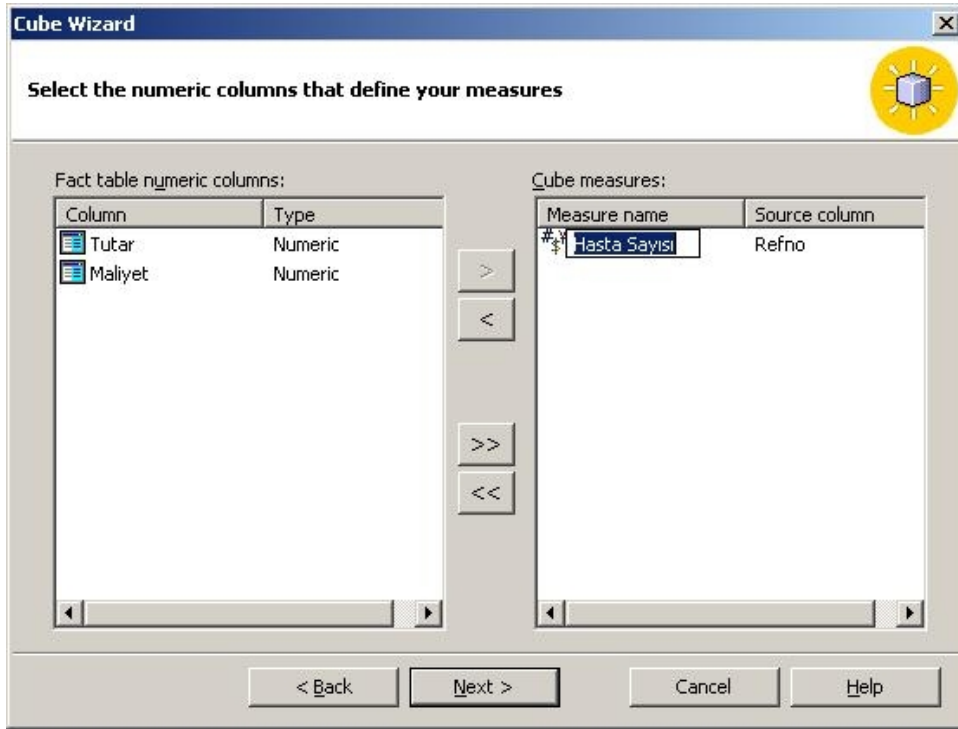


Şekil 5.7 New Cube Wizard Fact Table seçimi

Analysis Manager eğer daha önce yapılmamışsa tablonun konumlandığı data kaynağını tanımlamaya ve istenirse tablonun içindeki verilere göz atmaya olanak sağlar.

Fact Table'ın seçiminin ardından kullanıcı sihirbaz tarafından measure'ların seçimine yönlendirilir. Burada measure'ların özelliklerinden dolayı sadece nümerik

alanlar görülebilir. Measure'lar kullanıcının analiz etmek istediği nümerik değerlerdir. Measure'lara örnek olarak verilen hizmet tutarı değerini ya da bakılan hasta sayısını verebiliriz. Measure bilgileri veri kaynağındaki "fact table" dan alınır. Her küp en az bir measure içermelidir, ancak 1024 ten fazlasını içeremez. Şekil 5.8'de New Cube Wizard Measure seçimi gösterilmiştir.



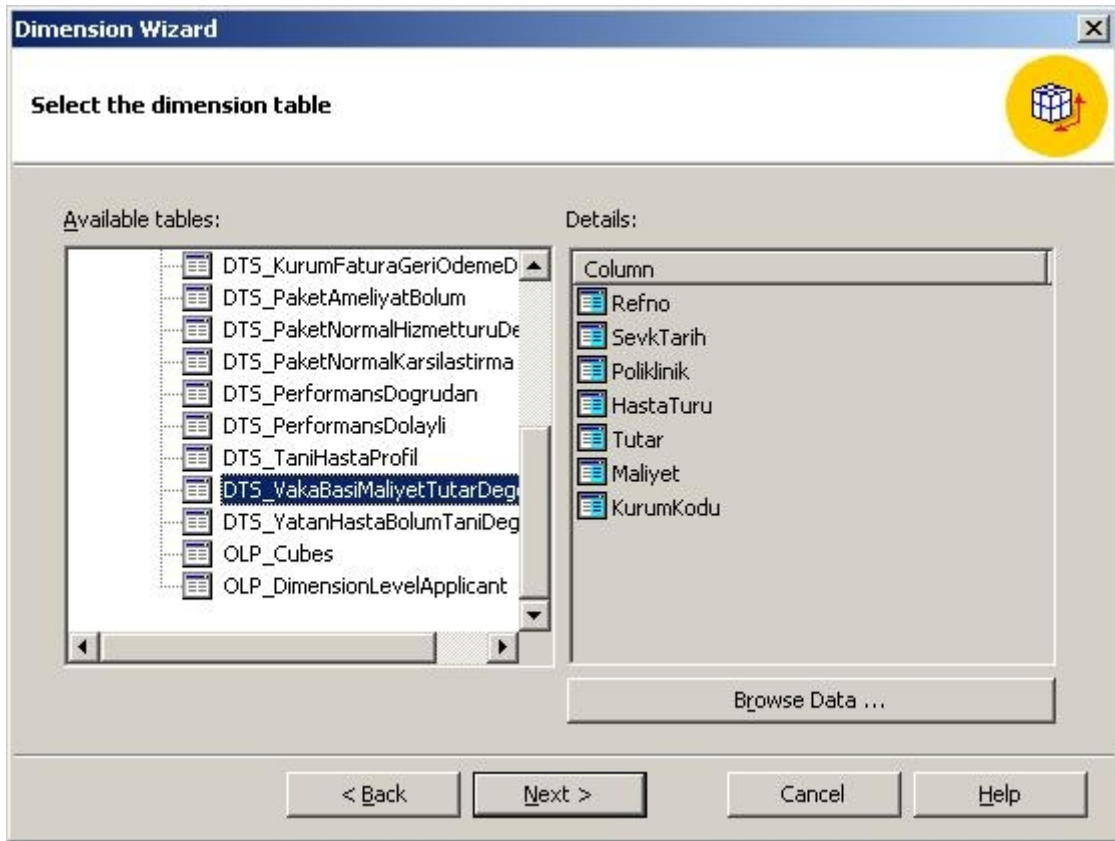
Şekil 5.8 New Cube Wizard Measure seçimi

Sihirbazda küpün Fact Table ve measure nesnelere belirlendikten sonra bir sonraki adım dimension'ların tanımlama ekranıdır. Dimension'ları level ve member'lardan oluşan veri analiz kategorileri olarak tanımlayabiliriz. Küpün birincil tablosundaki (Fact Table) verileri tanımlayarak level ve kategori hiyerarşisini düzenler. Bu kategori ve level'lar kullanıcının analiz etmek istediği benzer üye setlerini tanımlar. Dimensionlar hiyerarşiktir ve üyeleri çoğu kez piramide benzer bir konfigürasyonla düzenlenmiştir. Yatay yerleşim aynı level'daki sütun değerlerinden, dikey yerleşim ise farklı level lardaki sütun değerlerinden oluşur.

Level'lar, ortak bir anlam oluşturan member'ların gruplarıdır ve dimension tablo sütunlarını işaret ederler. Member'lar ise dimension'daki her bir ayrı değerdir. Dimension'lar, level'ları ve memberları ağaç yapısında organize eder.

Dimension seçimi yapılırken Fact Table'daki dimension nesnesini tanımlayacak, ayrıntılarını belirtecek bir Dimension Table seçilebilir. Fact Table'ın Dimension Key'i ile Dimension Table'daki Primary Key eşleniktir. Örneğin, Kurum Hastaları küpünün Fact Table'ı içerisindeki Kurum Kodu dimension'ı, bir Kurum Tanım Dimension Table'mın Kurum Kodu primary key'i ile ilişkilendirilerek kurum adı ve lokasyonu gibi bilgiler kullanıcıya sunulabilir.

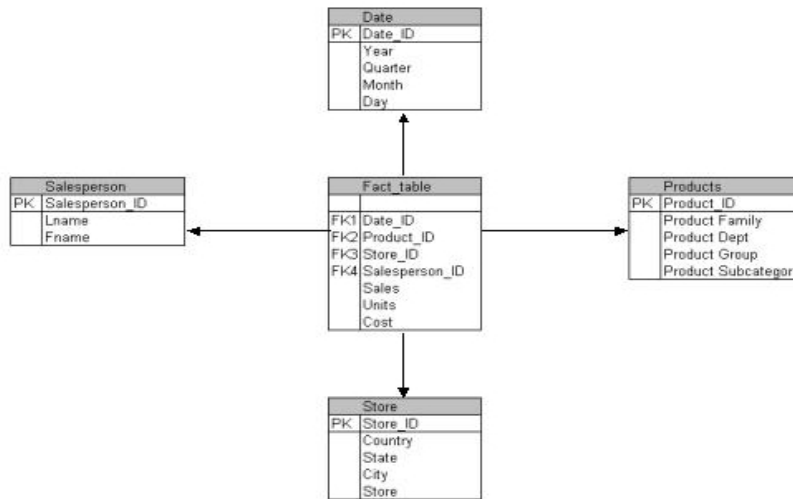
Dimension Tables: Dimension Table'lar Fact Table kayıtlarının özelliklerini açıklar. Bu özelliklerden bazıları açıklayıcı bilgiler sağlarken, diğerleri kullanıcıya yararlı bilgiler sağlamak için Fact Table verilerinin nasıl özetleneceğini belirtirler. Dimension table'lar member isimlerini, hiyerarşi tanımlarını ve diğer özelliklerini içerirler. Şekil 5.9'da Dimension Table seçim ekranı gösterilmiştir.



Şekil 5.9 Dimension Table seçim ekranı

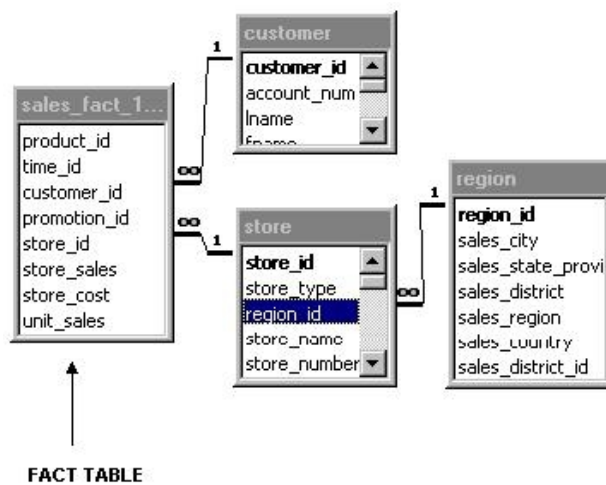
Dimension hiyerarşisi beş farklı şekilde olabilir. En çok kullanılan iki yöntem şunlardır:

Star Schema: Bir veya birden fazla sütunun bir tablodan seçildiği durumdur. Her sütun dimension'a bir level olarak katkıda bulunur. Eğer sütun tek tarih alanıysa birkaç level oluşturulabilir. Birden fazla sütun gittikçe artan detayda bilgiye sahip olmak için seçilir. Örneğin Ülke → İl → İlçe → Köy şeklinde bir dallanma seçilebilir. Şekil 5.10'da bir Star şema örneği gösterilmiştir.



Şekil 5.10 Star schema (Microsoft Corporation 2000)

Snowflake Schema: Bir veya birden fazla sütunun birden fazla ilişkili tablodan seçildiği durumdur. Her sütun dimension'a bir level olarak katkıda bulunur. Burada da birden fazla sütun gittikçe artan detayda bilgiyi göstermek amacıyla seçilir. Snowflake schema normalize edilmiş star schema olarak düşünülebilir. Şekil 5.11'de bir Snowflake şema örneği gösterilmiştir.

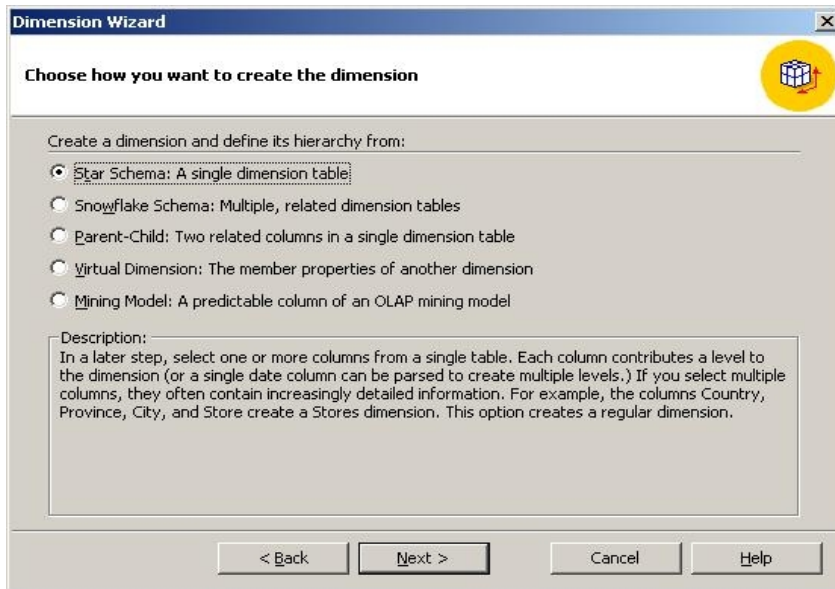


Şekil 5.11 Snowflake schema (Microsoft Corporation 2000)

Diğer dimension hiyerarşileri aşağıda listelenmiştir;

- Parent-Child,
- Virtual Dimension,
- Mining Model.

Dimension Table'ındaki birincil anahtar sütunu her bir dimension member'ı için benzersiz değer içermelidir. Her bir Dimension Table'ın birincil anahtarı ilişkili Fact Table'ın anahtar sütunlarından biri ile uyuşmalıdır. Dimension Table'da bir kez görünen her bir anahtar değer Fact Table'da ise bir çok kez görülebilir. Şekil 5.12'de Dimension hiyerarşisi seçim ekranı gösterilmiştir.



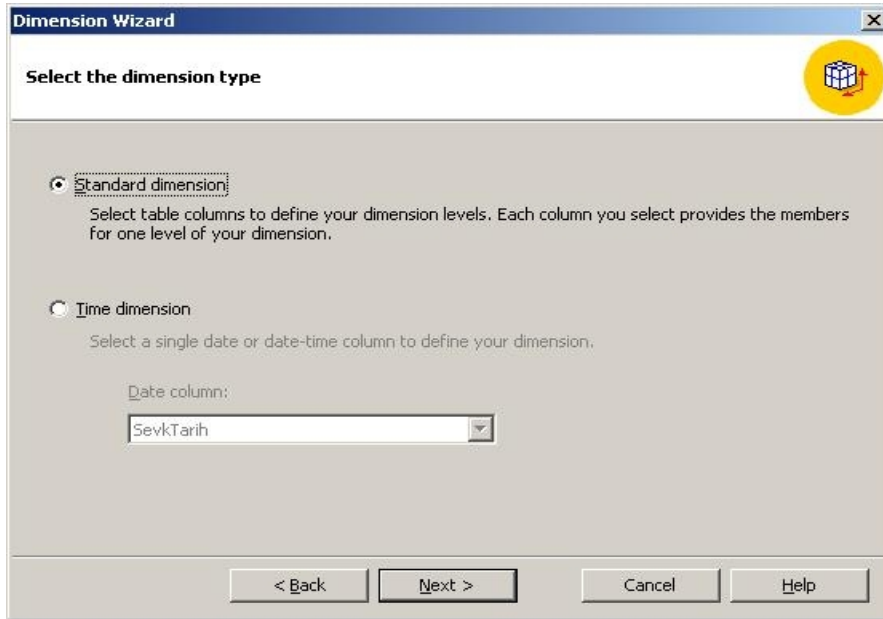
Şekil 5.12 Dimension hiyerarşi seçim ekranı

Dördüncü bölümde anlatıldığı üzere, istenen bilgiler değişik ilişkisel HBYS tablolarından iç içe sorgularla çekildiği için, çalışmada ayrıntılı bir dimension hiyerarşisine ihtiyaç duyulmayarak Star schema kullanılmıştır.

Dimension Hiyerarşisi ve Table'ının seçimi yapıldıktan sonra sıra dimension alanını belirlemeye gelir. Kullanıcıya dimension tipine göre iki seçenek sunulur:

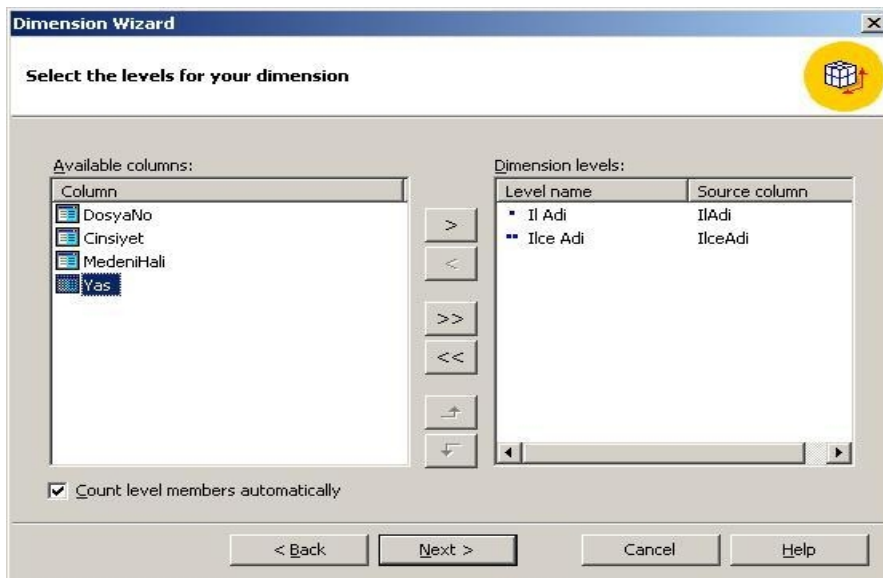
- Standart Dimension,
- Time Dimension.

Şekil 5.13'te Dimension tipi seçim ekranı gösterilmiştir.



Şekil 5.13 Dimension tipi seçim ekranı

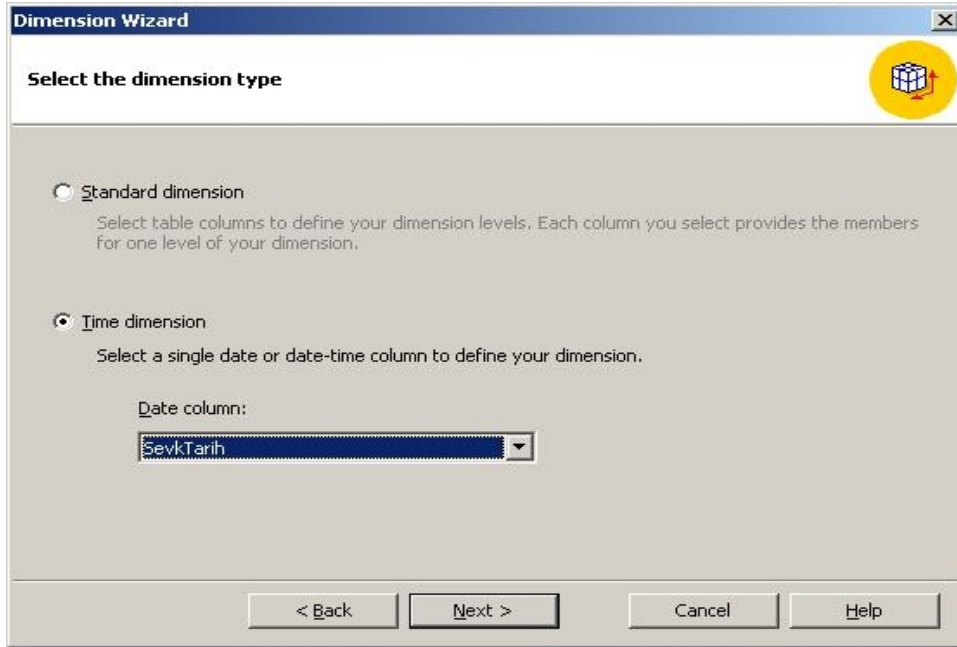
Seçim yapılırken istenirse birden çok level tanımı yapılarak dimension küp kullanıcısına değişik seviyelerde gösterilebilir. Çalışmada buna benzer bir örnek Şekil 5.14'te, Hasta Kimlik Bilgileri küpünün Doğum Yeri dimension'ı il ve ilçe şeklinde iki seviyede gösterilerek yapılmıştır.



Şekil 5.14 Standart Dimension Level seçimi

Time Dimension, ikinci bölümde anlatıldığı üzere OLAP teknolojisinin zaman yönelimli süreç kabiliyetini temsil eder. Küpün değişik zaman seviyelerinde

dilimlerinin alınmasına, measure bilgilerinin hesaplanmasına imkan tanır. Dimension Sihirbazı, Time Dimension seçiminin Fact Table'daki datetime özelliğine sahip alanlar arasından yapılmasına imkan tanır. Şekil 5.15'te Time Dimension seçimi gösterilmiştir.

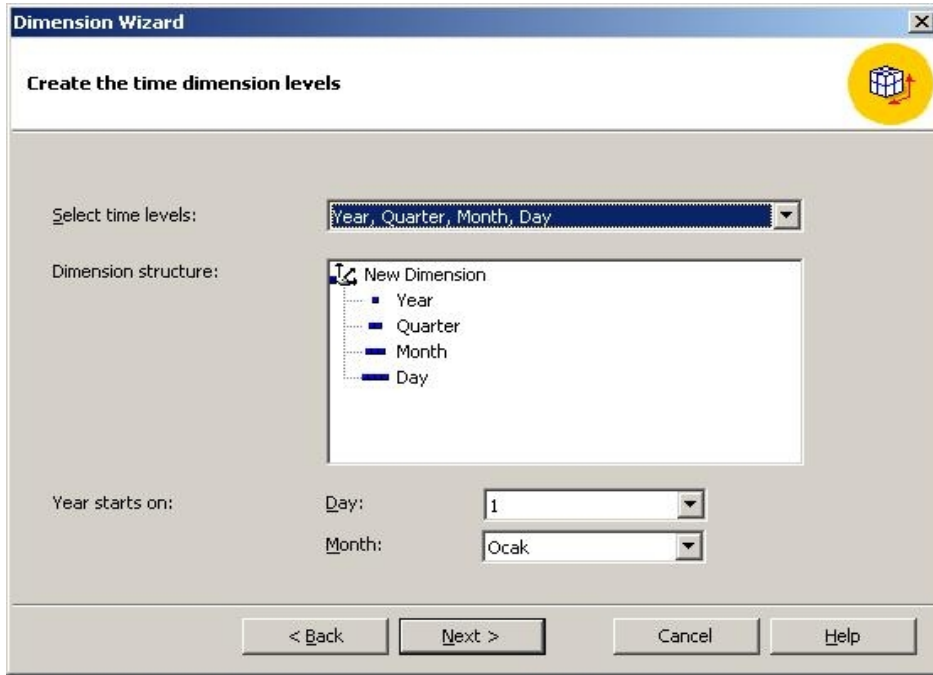


Şekil 5.15 Time Dimension seçimi

Dimension Sihirbazı, Time Dimension'ı kullanıcıya farklı yapılarla sunar:

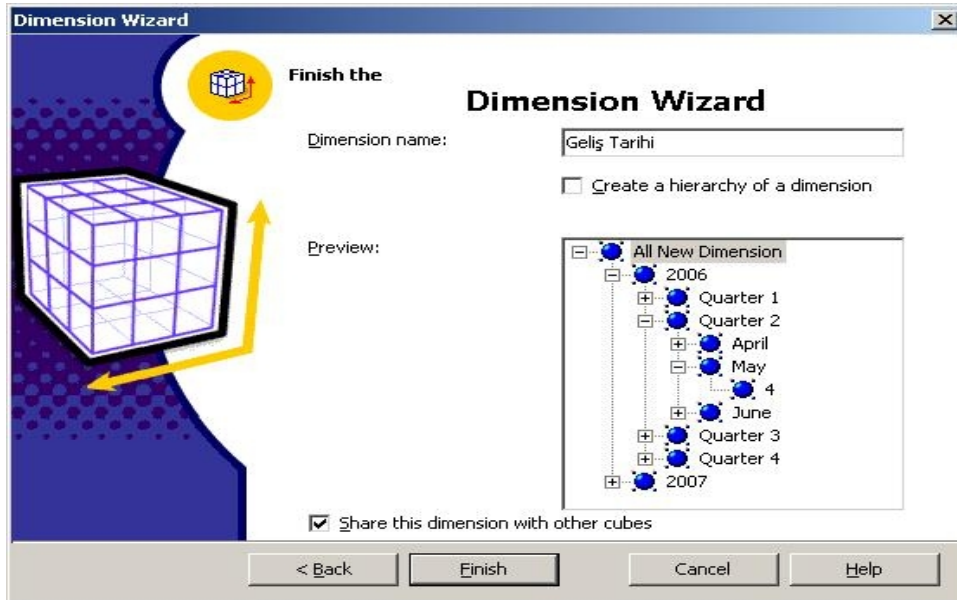
- Year, Quarter, Month, Day
- Year, Quarter, Month, Day, Hour, Minute
- Year, Quarter, Month
- Year, Month, Day
- Year, Month, Day, Hour, Minute
- Year, Month
- Year, Week, Day
- Year, Week, Day, Hour, Minute

Çalışmada Year, Quarter, Month ve Day seçeneği kullanılmıştır. Şekil 5.16'da Time Dimension Level seçimi gösterilmiştir.



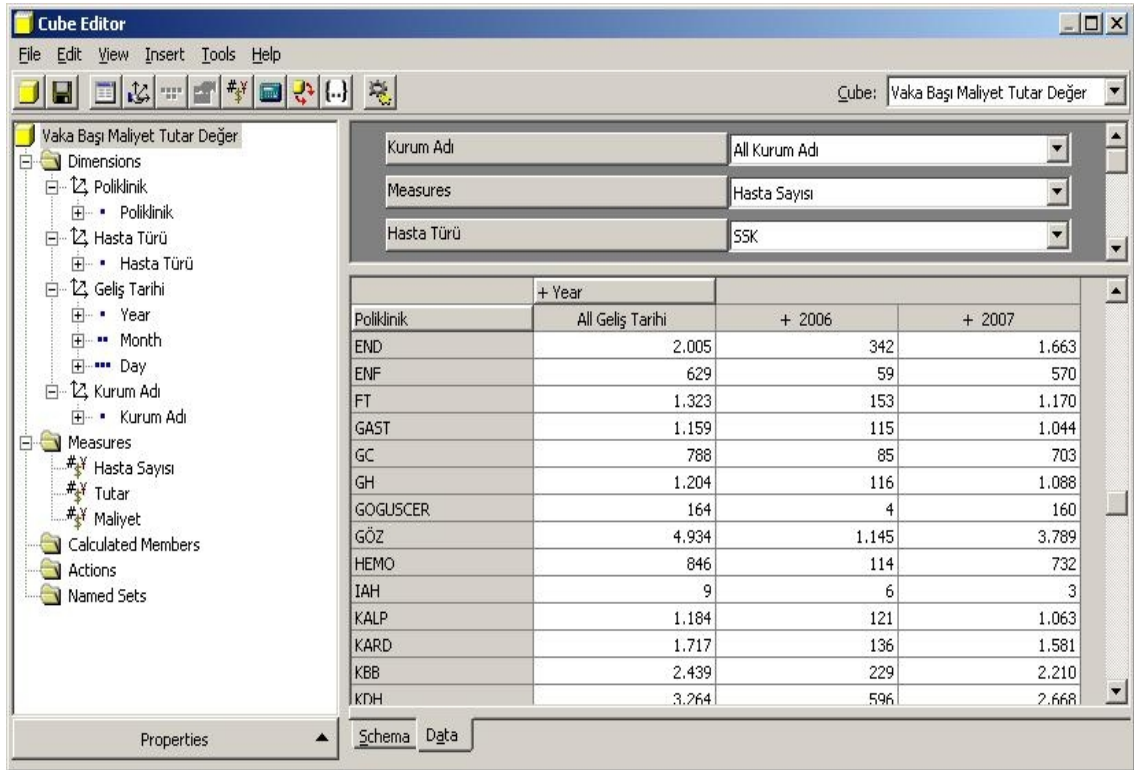
Şekil 5.16 Time Dimension Level seçimi

Kullanıcı, tanımını tamandıktan sonra dimension'ı diğer küplerde de kullanmak istiyorsa paylaşırabilir. Şekil 5.17'de Dimension Wizard'ın son ekranı gösterilmiştir.



Şekil 5.17 Dimension sihirbazı tamamlama ekranı

Dimension'ların oluşturulmasının ardından küp sihirbazı tamamlanmış olur. Şekil 5.18'de Cube Editor'de bir küpün yapısı ve veri görüntüsü yer almaktadır.



Şekil 5.18 Cube Editör

5.4 Process

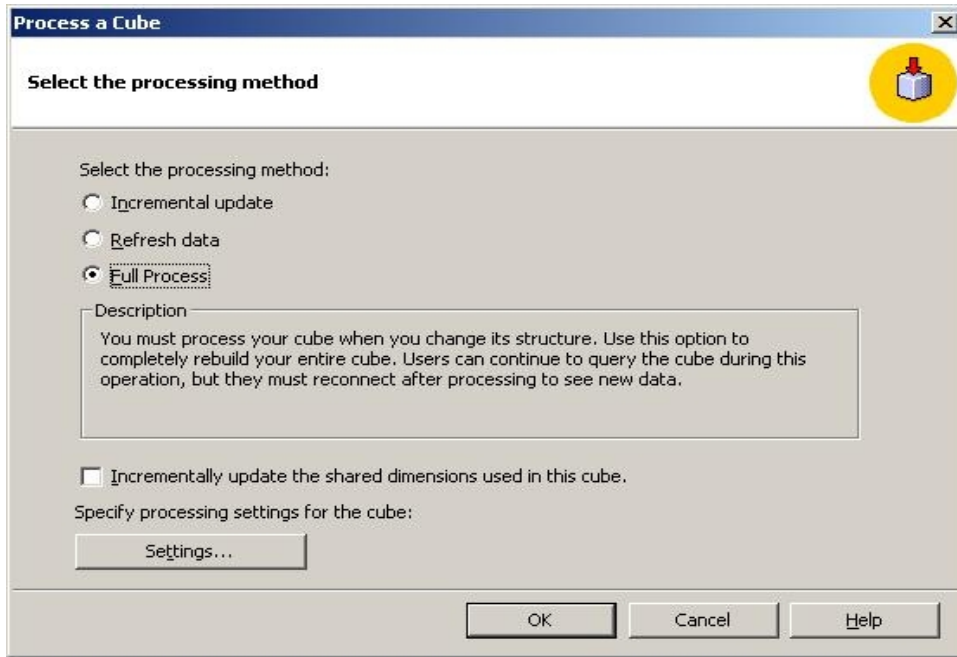
Küpler tasarlandıktan sonra oluşturulabilmesi için process komutunun yürütülmesi gerekir. Komut Analysis Manager'da seçili küpün üzerinde sağ tıklayarak verilir. Komut verildiğinde Analysis Server küp veri kaynağı ile bağlantıya geçerek verilerin son haline ulaşır. Process işlemi 3 farklı şekilde yapılabilir.

Incremental Update: Sadece veri kaynağındaki yeni datayı alır ve küp içinde konumlandırır. Process metotları içerisinde en az zaman alan metottur. İşlem tamamlandığında kullanıcılar yeni datalara ulaşırlar.

Refresh Data: Küp içerisindeki tüm datayı siler ve yeniden konumlandırır. Measure'lar yeniden hesaplanır. Küp yapısı değişmez. Bu işlem sırasında kullanıcılar kesinti olmaksızın küp sorgusu yapabilirler. Refresh işlemi bittiğinde yeni datalar görülür.

Full Process: Process metotları arasında en fazla zaman alan metottur. Varolan küpün silinip yeniden oluşturulması durumudur. Küpün dimension, measure yapısı değişikliklerini algılar. Küp ilk defa oluşturulduğunda full process yapılmak zorundadır.

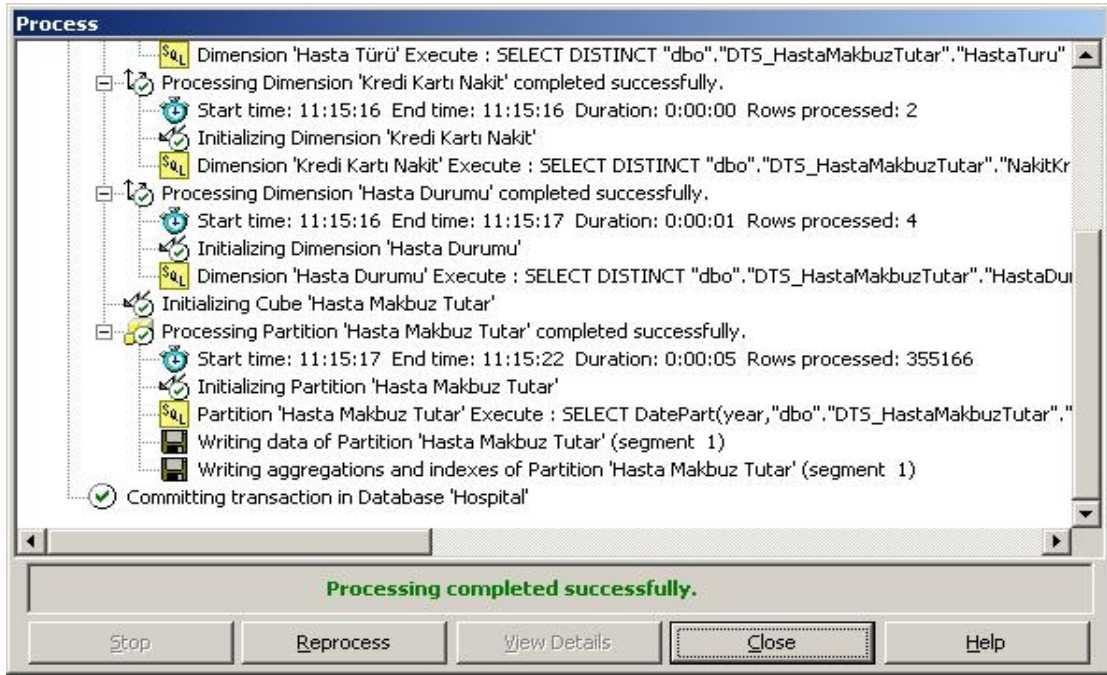
Şekil 5.19'da Process metodları gösterilmiştir.



Şekil 5.19 Process metodları

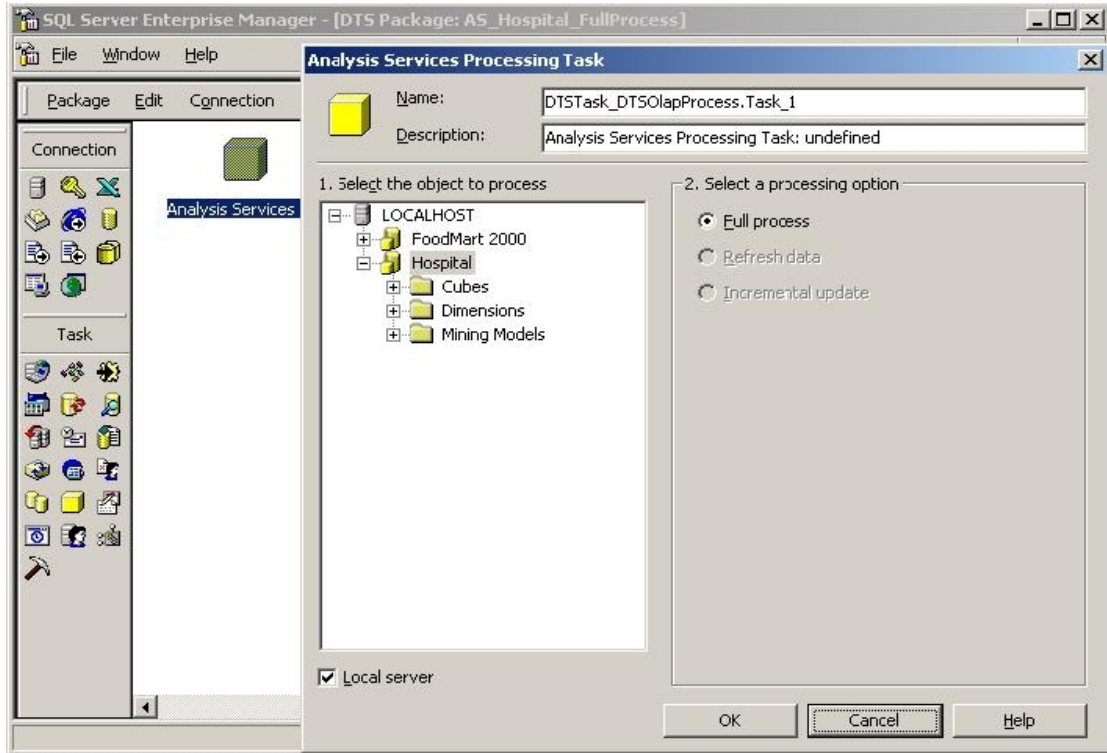
DTS ve Query Analyzer bölümlerinde de anlatıldığı üzere çalışmada güncel verilerin veri ambarına transferi, günlük çalışma saatleri dışında haftada bir, gece saatlerinde DTS paketlerinin zamanlanmasıyla yapılmaktadır. Veri ambarına taşınan güncel verilerin küp sorgularında görünebilmesi için process işleminin yapılması gereklidir.

Şekil 5.20'de Hospital analiz veritabanı üzerinde yapılan Full Process işlemi gösterilmiştir. Çalışmada, analiz veritabanı Hospital'ın Full Process yapılması bir DTS paketi haline getirilmiş ve sabah saat 05:00'te çalışacak şekilde ayarlanmıştır. Bu işlemin gerçekleştirilmesindeki amaç, veri taşıma DTS paketlerinin çalışması sonucu taşıdıkları yeni verilerin analiz veritabanına eklenmesidir.



Şekil 5.20 Process işlemi

Şekil 5.21’de Hospital Analiz Veritabanına Process Komutunu veren DTS Paketi gösterilmiştir.



Şekil 5.21 Hospital Analiz Veritabanı Process DTS paketi

Ancak bu DTS paketi, diđer veri taşıma paketleri gibi HBYS veritabanına deđil, analize kaynak oluřturan HospitalTEMP veri ambarına eklenmiř ve veritabanı bakım iřlemleri ile birlikte zamanlanmıřtır.

Bu iřlemlerden sonra OLAP küpleri kullanıma hazır hale gelmiřtir.

6. OLAPPROG YAZILIMI

OLAPPROG, Pamukkale Üniversitesi Hastaneleri HBYS için geliştirilmiş Microsoft Analysis Services OLAP küplerini kullanan bir yazılımdır.

OLAP küplerinin oluşturulması ve örnek veri görüntülerinin alınmasından sonra kullanıcıların küpleri rahatça kullanabilmesi, küp açıklamalarına ulaşabilmesi, grafik ve rapor desteği alabilmesi ve görünümde değişiklik yapabilmesi amacıyla bir yazılım geliştirilmiştir. OLAPPROG adıyla tanıtılacak bu yazılım, daha önce oluşturulmuş ve HospitalTEMP veritabanına kaydı yapılmış Microsoft Analysis Services küpleri ile bağlantı kurarak kullanıcıya detaylı analiz yapma imkanı sağlar. Program Pamukkale Üniversitesi Hastaneleri HBYS'nin içerisine entegre edilmiş olup yetkili kullanıcıların ulaşabileceği bir modül olarak çalışmaktadır.

Geliştirilen OLAPPROG yazılımı 5 alt bölümden oluşmaktadır:

1) Küp görüntüsü seçimlerinin yapılması

- Küp bilgisi ve küpe bağlanma
- Dimension Layout
- Measure Properties

2) Küp görünütüsü üzerinde çalışma

- Sorting
- Filtering
- Grouping

3) Grafik Desteği

4) Dosya olarak kaydetme - Çıktı alma

5) Ayarlar

- Görsel Ayarlar
- Veri Alanı Ayarları
- Alanları Ters Çevir
- Düzen

Şekil 6.1’de OLAPPROG yazılımının genel bir görüntüsü verilmiştir.

OLAP Rapor Ekranı

BAĞLAN DÜZEN AKTAR ÇIKTI AYARLAR GRAFİK KAPAT

Bağlanılan Rapor : Hasta Durumuna göre Gelir Hesabı

HASTANE BİLGİ YÖNETİM SİSTEMİ

KARAR DESTEK RAPORLARI

Hasta Kimlik Bilgileri

PAKET BİLGİLERİ

Paket Normal Karşılatma

Hizmet Türü Bazında Paket Normal Karşılatması

Paket Ameliyat Bilgileri

Hasta Durumuna göre Gelir Hesabı

Keşilen Fatura ve Geni Ödeme Bilgisi

TANI BİLGİLERİ

Yatan Hasta Tanı Bilgileri

Ayaktan Hasta Tanı Bilgileri

Tanı Hasta Profili

HİZMET BİLGİLERİ

Hizmet Türüne Göre İşlem Bilgileri

Hizmet Türüne Göre Step Değerleri

Hasta Türüne Göre Faturalama Bilgileri

MAKBUZ VE KATKI PAYI BİLGİLERİ

Hasta Türüne Göre Katkı Payı Bilgisi

Uzman Doktora Göre Katkı Payı Bilgisi

Hasta Makbuz Bilgileri

Vaka Baş Normal Karşılatma

PERFORMANS BİLGİLERİ

İsmi Giriş Olan İşlemlerin Performans Bilgileri

Konsültan Onay İşlemlerin Performans Bilgileri

5.Hasta Durumuna ve Bölümüne göre
a.Hastaların Sayı,
b.Venilen Hizmetin Tutarı,
c.Alınan Paramın Tutarı, dağılımı
Hastaların durumlarına ve hizmet aldıkları bölümlere göre
venilen hizmet tutarlarının makbuz ve fatura toplamı olarak
çıkartılması sağlanır. Venilen Hizmet Tutarı, Alınan Karşılık,
değerlerinin takip edilmesini kolaylaştıran geniş çaplı bir
rapordur. Bu raporda bölümlerin hasta sayısı ve tutar bazında
performansları ölçülebilir. Kesirlerinin nedeni ayrıntı olarak
performanslara yansıtılabilir.

Measures

Alınan

Hasta Sayısı

Hizmet Tutarı

Bölüm

Durum

Geliş Tarihi

Hasta Türü

Yatış Süresi

(Page area)

Bölüm

Geliş Tarihi

Measures

(Column area)

Hasta Türü

Hasta Sayısı

Hizmet Tutarı

(Measure area)

Year

% 2006

% 2007

% 2008

Total

Measures

Hasta Sayısı

Hizmet Tutarı

Hasta Sayısı

Hizmet Tutarı

Hasta Sayısı

Hizmet Tutarı

Hasta Sayısı

Hizmet Tutarı

(Row area)

Bölüm	Hasta Sayısı	Hizmet Tutarı	Hasta Sayısı	Hizmet Tutarı	Hasta Sayısı	Hizmet Tutarı	Hasta Sayısı	Hizmet Tutarı
+ GÜZ	24.898	2.824.833,82	25.850	2.211.220,67	5.127	221.435,39	55.875,00	5.257.489,88
+ KDH	15.571	2.799.565,42	17.002	2.679.620,60	5.070	708.260,75	37.643,00	6.187.446,77
+ END	13.938	1.384.453,66	16.620	1.447.192,02	4.776	396.629,08	35.334,00	3.228.274,76
+ ACIL	10.094	964.831,63	18.413	1.596.762,52	6.448	539.302,67	34.955,00	3.100.896,82
+ KARD	13.560	2.338.232,89	13.406	2.635.445,36	4.499	818.453,80	31.465,00	5.792.132,05
+ PSK	13.572	751.105,06	14.102	697.955,20	3.759	139.806,11	31.433,00	1.588.866,37
+ KBB	10.150	1.145.985,96	12.741	1.334.495,48	3.594	281.729,12	26.485,00	2.762.210,56
+ ORT	8.064	1.739.941,73	12.156	1.823.994,66	3.135	502.089,03	23.955,00	4.066.025,42
+ NDR	7.562	797.299,54	10.607	1.061.967,69	2.740	288.138,66	20.909,00	2.147.405,89
+ DERM	7.480	414.947,83	9.941	518.182,96	2.877	127.057,43	20.298,00	1.060.188,22
+ ÜRD	6.739	1.248.621,68	9.565	1.638.313,01	2.664	450.412,34	18.988,00	3.337.347,03
+ FT	5.539	931.969,24	7.977	1.097.338,37	2.404	242.918,93	15.920,00	2.272.226,54
+ ÇH	6.533	1.052.925,57	6.104	842.741,08	2.686	222.403,21	15.323,00	2.118.069,86
+ BC	4.424	2.313.702,22	8.098	3.017.169,43	2.420	911.352,13	14.942,00	6.242.223,78
+ GH	5.048	910.947,72	7.563	1.269.112,42	2.292	365.149,21	14.903,00	2.545.209,35
+ DHL	4.873	337.863,36	6.115	437.337,67	1.444	98.883,71	12.432,00	873.894,74
+ GAST	4.141	994.736,97	5.861	1.397.020,91	1.895	381.890,16	11.887,00	2.773.648,04
+ ROM	3.796	438.652,16	5.631	619.193,50	1.551	151.778,77	10.978,00	1.209.624,43
+ ONKO	3.463	732.008,29	5.972	1.018.277,54	1.530	198.442,82	10.965,00	1.948.728,75
+ KALP	3.644	2.755.742,47	5.728	3.462.113,85	1.540	762.397,47	10.912,00	6.980.253,79
+ GC	3.779	1.978.202,05	5.359	2.282.038,93	1.680	659.199,95	10.818,00	4.919.440,93

Şekil 6.1 OLAPPROG yazılımı

6.1 Küp Görüntüsü Seçimlerinin Yapılması

6.1.1 Küp bilgisi ve küpe bağlanma

Kullanım için öncelikle küplerin Analiz Server’da oluşturulmuş ve tanımlanmalarının Analiz Server veritabanı OLP_Cubes tablosunda yapılmış olması gerekir. Tablodaki Sirano, ParentKod ve Kod alanları küplerin OLAPPROG içerisinde ağaç yapısında görünümünü sağlar. CubeName, Açıklama ve Özet alanları ise OLAPPROG ana

ekranında küp isimlerinin ve bilgilendirmelerinin görülebilmesi içindir. Şekil 6.2’de küp tanımlamalarının bulunduğu OLP_CUBES tablosu gösterilmiştir.

Sirano	ParentKod	Kod	CubeName	Açıklama	Özet	Cube
0	-1	0	YOK	KARAR DESTEK RAPORLARI	<NULL>	0
1	0	100	Hasta Kimlik Bilgileri	Hasta Kimlik Bilgileri	1.Hasta Kimlik Bilgileri	1
2	0	200	YOK	PAKET BİLGİLERİ	<NULL>	0
3	2	210	Paket Normal Karşılaştırma	Paket Normal Karşılaştırma	2.Paket Hastaların bölüme göre	1
4	2	220	Paket Normal Hizmet Türü Detay	Hizmet Türü Bazında Paket Normal Karşılaştırması	3.Pakete Dönüştürülen Referansların Bölüme Göre,	1
5	2	230	Paket Ameliyat Bölüm	Paket Ameliyat Bilgileri	4.Paket Ameliyatların bölüme göre (Paket Ameliyat Bölüm)	1
6	0	300	Hasta Durum Bölüm Gelir Dağılım	Hasta Durumuna göre Gelir Hesabı	5.Hasta Durumuna ve Bölümüne göre	1
7	0	400	Kurum Fatura Geri Ödeme Dağılım	Kesilen Fatura ve Geri Ödeme Bilgisi	6.Kurumlarına göre	1
8	0	500	YOK	TANI BİLGİLERİ	<NULL>	0
9	8	510	Yatan Hasta Bölüm Tanı Değer	Yatan Hasta Tanı Bilgileri	7.Yatan hastaların Tanı ve ABD na göre,	1
10	8	520	Ayaktan Hasta Bölüm Tanı Değer	Ayaktan Hasta Tanı Bilgileri	8.Ayaktan hastaların Tanı ve ABD na göre	1
11	8	530	Tanı Hasta Profili	Tanı Hasta Profili	9.Tanı Hasta Profili	1
12	0	600	YOK	HİZMET BİLGİLERİ	<NULL>	0
13	12	610	Hizmet Türü Bölüm İşlem Tutar Değer	Hizmet Türüne Göre İşlem Bilgileri	10.Hizmet türüne ve bölüme göre	1
14	12	620	Hizmet Türü Prosedür Değer	Hizmet Türüne Göre Step Değerleri	11.Tedavisi tamamlanmış hastaların uzman hekim onayı alınmamış Ra	1
15	0	700	Fatura Sosyal Güvence	Hasta Türüne Göre Faturalama Bilgileri	12.Hastaların Sosyal Güvencelerine göre Faturalama Değerleri	1
16	0	800	YOK	MAKBUZ VE KATKI PAYI BİLGİLERİ	<NULL>	0
17	16	810	Katki Payı Hasta Türü	Hasta Türüne Göre Katki Payı Bilgisi	13.Hasta Türü Katki Payı Analizi	1
18	16	820	Katki Payı Doktor	Uzman Doktora Göre Katki Payı Bilgisi	14.Uzman Doktor Katki Payı Analizi	1
19	16	830	Hasta Makbuz Tutar	Hasta Makbuz Bilgileri	15.Hasta Makbuz Bilgileri	1
20	0	900	Vaka Başı Maliyet Tutar Değer	Vaka Başı Normal Karşılaştırma	16.Vaka başı Hastaların bölüme göre	1
21	0	1000	YOK	PERFORMANS BİLGİLERİ	<NULL>	0
22	21	1010	Performans Doğrudan	İsim Girişi Olan İşlemlerin Performans Bilgileri	17.Bölüme Göre Öğretim Üyeleri Performans Uygulaması Analizi	1
23	21	1020	Performans Dolaylı	Konsültan Onaylı İşlemlerin Performans Bilgileri	18.Bölüme Göre Öğretim Üyeleri Performans Uygulaması Analizi	1

Şekil 6.2 OLP_CUBES tablosundan görünüm

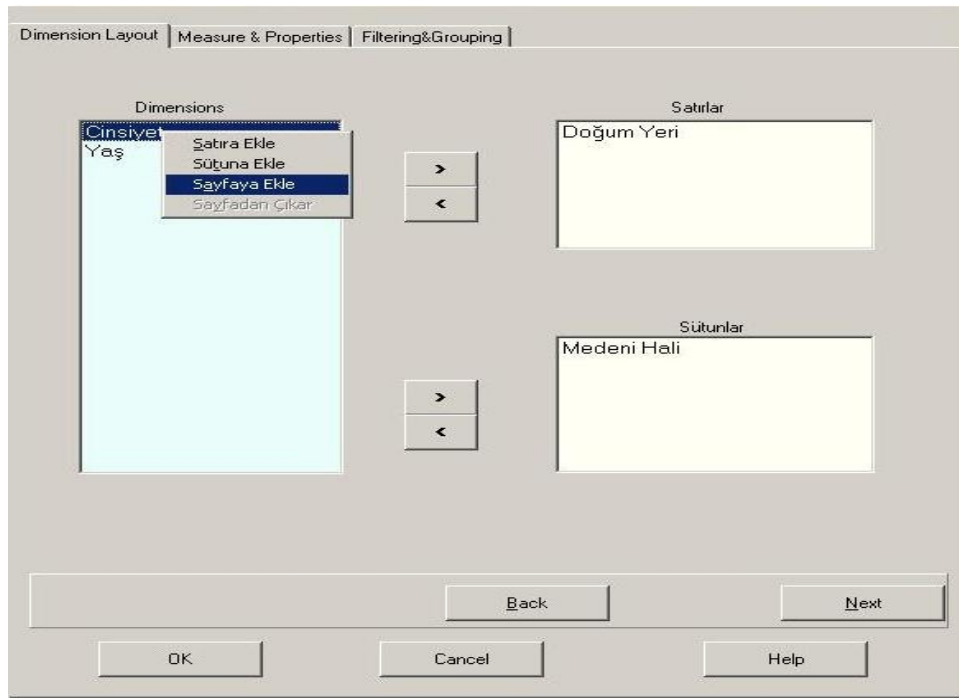
Tanımı yapılmış olan küpe bağlanmak için bağlan menüsünden küp seçilir ya da küp seçildikten sonra üzerinde sağ tıklanır. Bu işlem kullanıcıyı Küp Ayarları Seçim Ekranına götürür.

6.1.2 Dimension layout

Kullanıcı iki tablı seçim ekranının birincisinde Dimensionların satır, sütun ve sayfa yerleşimlerini yapabilir. Satır, sütun ve sayfa küpün üç boyutu gibi düşünülerek, yerleşim ona göre planlanabilir. Şekil 6.3’te küpteki dimension yerleşiminin yapılması gösterilmiştir.

6.1.3 Measure ve properties

Hesaplanması istenen ölçüm değerleri ikinci tabda seçilebilir. Burada sadece ölçümlerin normal değerleri değil aynı zamanda satır, sütun, ebeveyn ve genel toplama göre yüzde değerleri de alınabilir.



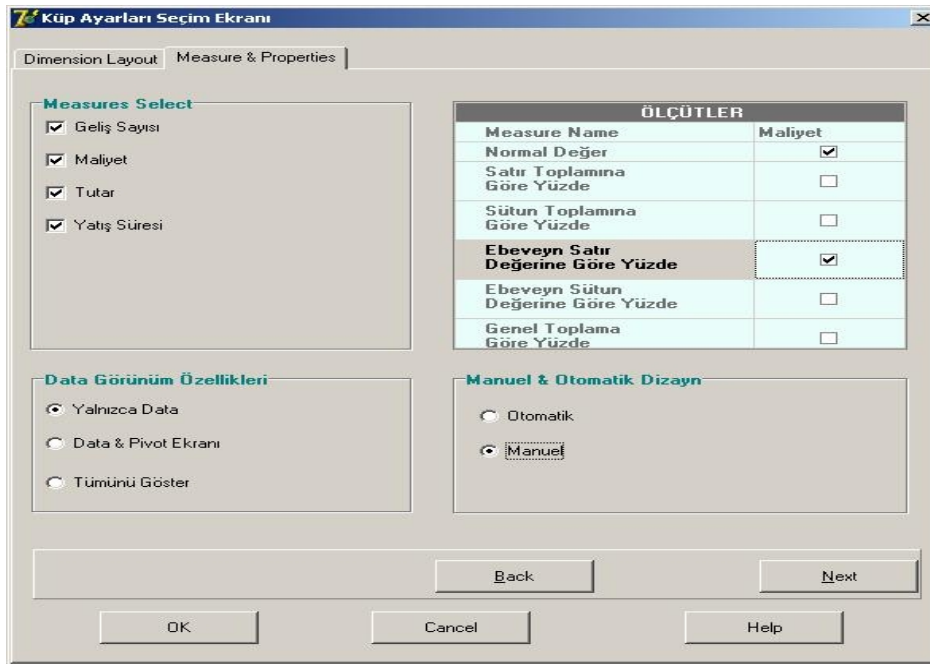
Şekil 6.3 Dimension yerleşimi

Özellikler kısmında ise küpün veri görünümü üç şekilde kullanıcının seçimine sunulur.

1. Yalnızca Data: Ana ekranda küpün gösterildiği gridde sadece verinin görünümü kullanıcıya sunulur.
2. Data & Pivot Ekranı: Bu seçenek kullanıcının seçili dimensionlar arasında pivotlama yapabilmesini sağlar.
3. Tümünü Göster: Seçili olmayan dimensionları ve measure'ları küp görünümüne ekleme, seçili olanları çıkarabilme ve pivotlama yeteneklerini kullanıcıya sunar.

Bu tabda, Küp ayarları seçim ekranında yapılan tüm işlemlerin daha sonra grid üzerinde değiştirilmesi amacıyla Manuel veya Otomatik dizayn seçeneği de verilmiştir. Şekil 6.4'te measure seçimi ve özellikleri gösterilmiştir.

Programda çok boyutlu küp görünümüne ulaşmak için Radar-Soft firması tarafından üretilen RadarCube VCL for Microsoft Analysis Services nesnelere kullanılmıştır.



Şekil 6.4 Measure seçimi

Radar-Soft, Business Intelligence konusunda yazılımcıların programlarında kullanabilmesi amacıyla değişik platformlarda nesne geliştiren bir firmadır. OLAPPROG'da kullanılan ve küp görünümünü kullanıcıya sunan OLAPGRID nesnesi RadarCube VCL ürününün bileşenidir.

6.2 Küp Görünütüsü Üzerinde Çalışma

6.2.1 Sorting

OLAPGRID'le küp görüntüsü üzerinde çalışma yaparken sağlanan özelliklerden biri de sıralama seçeneğidir. Sıralama, artan, azalan veya default değerinde member isimlerine göre veya measure değerlerine göre yapılabilir. Bu özellikler measure değerleri üzerinde sağ tıklandığında kolayca erişilebilecek şekilde tasarlanmıştır. Şekil 6.5'te measure değerlerine göre sıralama gösterilmiştir.

6.2.2 Filtering

Filtreleme özelliği kullanıcının küp görüntüsüne önem verdiği noktalara göre bakmasını sağlar. Çok üyeli dimensionlarda measure değerleri az olan member'lar göz ardı edilebilir. Gridde filtreleme özelliği hem dimension bazında Hierarchy Editor'de, hem de member bazında memberlar üzerinde sağ tıklanınca kullanılabilir.

OLAP Rapor Ekranı
BAĞLAN DÜZEN AKTAR ÇIKTI AYARLAR GRAFİK

Bağlanılan Rapor : Hasta Durumuna göre Gelir He:

Year	2006	2007	Total
Bölüm			
ACIL	10.094	17.299	27.393,00
PSK	13.572	13.560	27.132,00
KARD	13.560	12.743	26.303,00
KBB	10.150	12.200	22.350,00
ORT	8.064	11.658	19.722,00
NÖR	7.562	10.162	17.724,00
DERM	7.481	9.480	16.961,00
ÜRD	6.739	9.177	15.916,00
FT	5.539	7.629	13.168,00
ÇH	6.533	5.780	12.313,00
GH	5.048	7.196	12.244,00
BC	4.424	7.728	12.152,00
DHL	4.873	5.868	10.741,00
GAST	4.142	5.559	9.701,00
ROM	3.796	5.356	9.152,00
ONKO	3.464	5.686	9.150,00
KALP	3.644	5.502	9.146,00
GC	3.779	5.127	8.906,00
ÇA	3.405	4.622	8.027,00
HEMO	2.662	4.687	7.349,00
NTIP	3.289	2.767	6.056,00
ÇE	1.980	3.622	5.602,00
NEF	2.462	2.924	5.386,00
PC	1.641	3.372	5.013,00

Sort ascending
Sort descending
No sorting
Show as

Şekil 6.5 Sorting

Örneğin, bölümlere ve geliş tarihine göre hasta sayıları üzerinde çalıştığımızı düşünelim. Eğer 2006 yaz dönemi hasta sayısı ile 2007 yaz dönemi hasta sayılarını karşılaştırmak istiyorsak, Hierarchy Editor'den sadece 2006 ve 2007 yılları seçili bırakılır. 2006 ve 2007 yıllarında bir alt seviyeye inilerek ay bazında Haziran, Temmuz ve Ağustos ayları seçili bırakılır. İşlem onaylandığında sadece o aylara ait measure değerleri gelecektir. Şekil 6.6'da Hierarchy Editor'den filtreleme işleminin yapılması gösterilmiştir.

Filtreleme özelliği, sıralama ile birlikte kullanılınca istenen konuma daha kolay ulaşılabilir. Paket fatura tutarına göre en üst değere sahip 10 bölüm karşılaştırılmak istendiğini düşünelim. Önce gerekli dimension ve measure değerleri seçilir ve küp görüntüsü grid üzerinde alınır. Daha sonra grid üzerinde measure değerlerine göre sıralama yapılır. Bu işlem, yapılmak istenen member başlığına tıklanarak gerçekleştirilebilir veya değerler üzerine sağ tıklanarak yapılabilir. Member üyeleri sıralandıktan sonra filtreleme için değişik yollar denenebilir. Bunlar;

- Belli bir değer altındaki memberlar, üzerlerinde sağ tıklanarak gizlenebilir,
- Belli bir değer üstündeki memberlar, üzerlerinde sağ tıklanarak gizlenebilir,
- Üstten veya alttan herhangi bir sayıda member görüntülenebilir.

Bağlanılan Rapor : Hasta Durumuna göre Gelir Hesabı

Year	2006	2007	Total
Bölüm			
ACIL	10.094	17.299	27.393,00
ACILR		1	1,00
ADT	1.708	839	2.547,00
AĞRI	12	52	64,00
ANEST	570	557	1.127,00
ANESYOĞ	67	250	317,00
ASTIM	366	469	835,00
BC	4.424	7.728	12.152,00
BCYB	40	78	118,00
BIYD	1.765	579	2.344,00
Ç.H	994	78	1.072,00
ÇA	3.405	4.622	8.027,00
ÇAL	730	2.914	3.644,00
ÇC	1.755	2.864	4.619,00
ÇE	1.980	3.622	5.602,00
ÇH	6.533	5.780	12.313,00
ÇK	539	2.156	2.695,00
ÇN	1.222	2.309	3.531,00
ÇP	2.144	2.284	4.428,00
DERM	7.481	9.480	16.961,00

Hierarchy editor - Geliş Tarihi

- 2004
- 2005
- 2006
- 2007
 - January
 - February
 - March
 - April
 - May
 - June
 - July
 - August
 - September
 - October
 - November

OK Cancel

Şekil 6.6 Hierarchy Editor

Şekil 6.7’de member filtreleme işlemi gösterilmiştir.

Bağlanılan Rapor : Paket Normal Karşılaştırma

Hasta Türü	Total	Bağkurlu	Emekli	Sevкли	SSK	Ücretli	Yeşil Kart
Bölüm							
KALP	5.241.291,86	1.039.791,22	898.571,61	253.871,02	2.269.522,33	185.025,05	594.510,63
KARD			746.963,79	184.205,39	1.233.912,04	83.815,53	136.795,41
GÖZ			1.016.226,12	137.208,04	863.913,30	33.720,62	155.159,23
KDH			220.738,62	455.236,74	502.324,63	27.067,35	21.040,48
KBB				187.294,80	249.246,61	8.259,44	14.616,14
GC				134.731,46	159.232,27	7.035,43	9.978,41
ÜRD				89.860,15	212.710,18	1.642,81	25.116,27
ORT	444.250,67	18.401,76	10%	68.867,78	228.616,96	12.708,58	54.530,75
BC	262.327,40	11.269,88	25%	47.617,32	121.711,36	15.353,63	17.827,05
ÇC	256.548,09	14.585,49	50%	83.028,06	121.648,73	3.427,09	26.985,12
Total	14.323.272,19	1.875.966,57	75%	1.641.920,76	5.962.838,41	378.055,53	1.056.559,49
			Other...				

Hide this member
Hide all members except this
Hide all members below this
Show only the top 1
Show only the bottom 3
Show all members 5
Create new group...

Şekil 6.7 Member filtreleme

6.2.3 Grouping

Kullanıcılar çalıştıkları küp üzerinde memberları belirli gruplar altında toplu olarak görmek isteyebilirler. Bu özellik OLAPGRID üzerinde sağ tıklanıp, create new group seçeneği seçilerek gerçekleştirilebilir. Bu işlemi bir örnek üzerinde açıklayalım. Doğum yerlerine göre hasta sayıları bölgesel olarak karşılaştırılmak istenirse grup oluşturulabilir. Şekil 6.8’de bu örneğe göre member gruplama gösterilmiştir.

Bağlanılan Rapor : Hasta Kimlik Bilgileri						
Medeni Hali		Bekar	Boşanmış	Dul	Evli	Total
İl Adı						
EGE	2.005,00	69.998,00	38,00	3.030,00	122.969,00	198.040,00
AFYONKARAHİSAR	88	2.634	5	156	7.895	10.778
AYDIN	69	2.169	2	108	4.377	6.725
BURDUR	77	1.508	3	142	5.743	7.473
DENİZLİ	1.576	57.541	25	2.384	94.421	155.947
ISPARTA	20	612		29	1.230	1.891
İZMİR	68	2.171		40	1.658	3.937
KÜTAHYA	5	301		16	578	900
MANISA	33	1.093	1	42	2.061	3.230
MUĞLA	25	925	1	42	2.066	3.059
UŞAK	44	1.044	1	71	2.940	4.100
DİĞER	519,00	15.744,00	9,00	390,00	18.710,00	35.372,00
Total	2.524	85.742	47	3.420	141.679	233.412

Şekil 6.8 Member gruplama

Programda yeni grup oluşturma, gruba member ekleme, gruptan member çıkarma, grup içerisindeki elemanları boşaltma, grup silme, grubu yeniden adlandırma gibi özellikler mevcuttur. OLAPPROG hem gruplama hem filtreleme özelliğini desteklemez. Şekil 6.9’da gruba member atama işlemi gösterilmiştir.

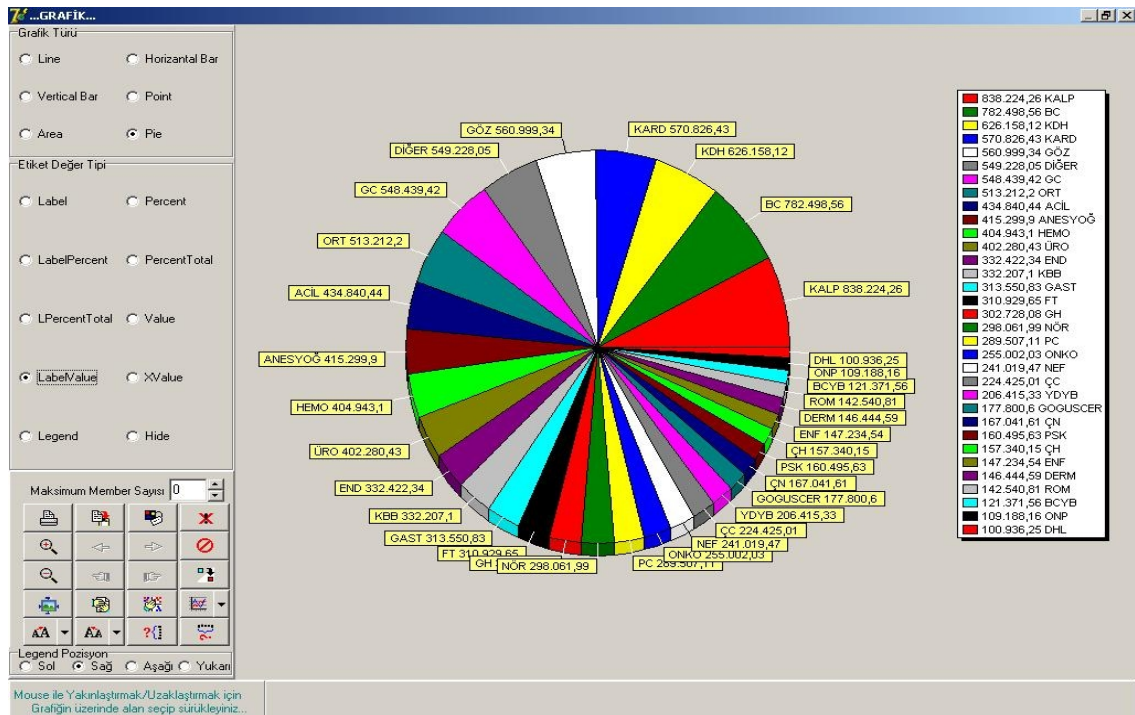
6.3 Grafik Desteği

OLAPPROG kullanıcıyı grafiksel olarak da destekler. 2007 yaz dönemi bölümler arası hizmet toplamını araştırdığımızı düşünelim. Küp görüntüsü birinci ve ikinci maddede anlatıldığı üzere OLAPGRID üzerinde oluşturulur. Bu noktada menüden grafik seçilmeden önce kullanıcıdan seçimini özelleştirmesi, istediği verileri filtrelemesi beklenir.

Bağlanılan Rapor : Hasta Kimlik Bilgileri						
Medeni Hali	Bekar	Boşanmış	Dul	Evli	Total	
AYDIN	69	2.169	2	108	4.377	6.725
AZERBAIJAN		1			1	2
BALIKESİR	10	220		14	523	867
BARTIN						44
BATMAN						63
BAYBURT						35
BILECİK						60
BİNGÖL						65
BITLİS						22
BOLU						77
BULGARİSTAN						
BURDUR						
BURSA						
ÇANAKKALE						
ÇANKIRI	2	54		2		
ÇORUM	9	194	1	9		
DENİZLİ	1.576	57.541	25	2.384	94.421	135.347

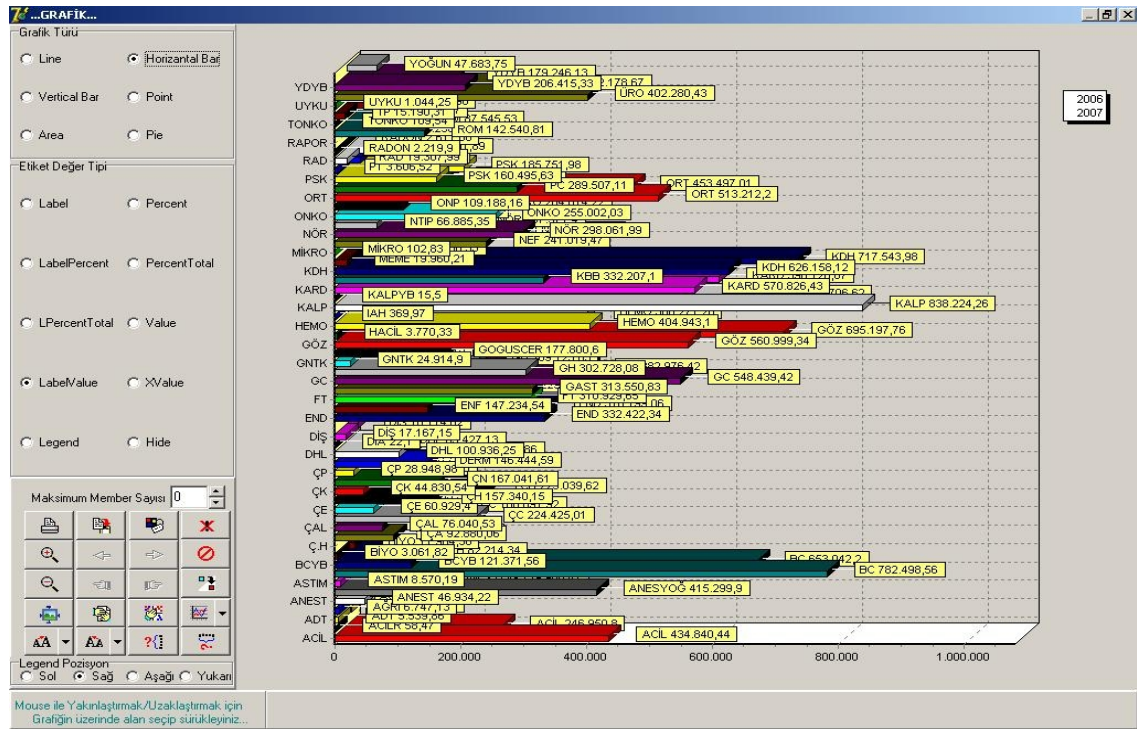
Şekil 6.9 Gruba Member atama

Şekil 6.10'da veriler üzerinde önce sıralama yapılmış, daha sonra 2007 yaz döneminde verdiği hizmet tutarı 100.000 YTL'nin altında olan bölümler oluşturulan "DİĞER" grubunun içine dahil edilmiştir. Bu işlemler sonucunda grafik daha anlaşılır hale gelmiştir.



Şekil 6.10 Grafik –tek Dimension ve tek Measure

Grafik ekranı 6 farklı görünümü ve 10 ayrı etiket türünü destekler. Pie görünümü haricinde grafikteki maksimum member sayısı belirlenebilir ve sayfalar arasında geçiş yapılabilir. Grafik özellikleri üzerinde değişiklik yapılabileceği gibi dosya olarak veya yazıcıdan çıktı alınabilir. Grafik ekranı birden fazla measure desteklemez ve yoğun verilerle kullanılması tavsiye edilmez. Şekil 6.11'de yoğun verili ve iki Dimension'lı grafik görünümü gösterilmiştir.



Şekil 6.11 Yoğun verili ve iki Dimension'lı grafik görünümü

6.4 Dosya Olarak Kaydetme-Çıktı Alma

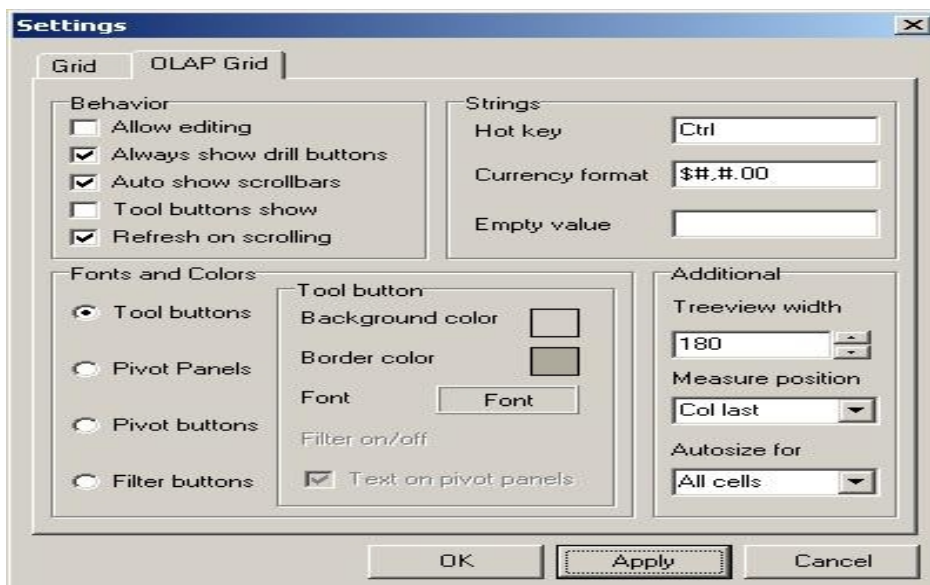
OLAPPROG ekranda görülen bilgilerin yazıcıdan çıktısını alabilmeyi ve dosya olarak kaydetme özelliğine de sahiptir. Alınan yıllar arası gelir raporu küpünün tamamı ya da belirli bir dilimi Excel belgesi olarak saklanarak, konuyla ilgili diğer çalışanlara iletebilir. OLAPPROG TXT, HTML, RTF, PDF, XML, BMP, WMF, XLS uzantılı dosya çıktısı üretebilir. Ayrıca ister seçili alan, istenirse OLAPGRID'in tümü yazdırılabilir. Bu işlemler ana ekrandaki Çıktı ve Aktar menülerinden gerçekleştirilebilir. Şekil 6.12'de OLAPPROG'dan alınan Excel belgesi çıktısı gösterilmiştir.

Year	2006		2007		Total	
Bölüm	Hasta Sayısı	Hizmet Tutarı	Hasta Sayısı	Hizmet Tutarı	Hasta Sayısı	Hizmet Tutarı
ACIL	10.094,00	964.831,63	17.299,00	1.513.353,68	27.393,00	2.478.185,31
ADT	1.708,00	31.890,46	839,00	16.159,96	2.547,00	48.050,42
AĞRI	12,00	2.847,88	52,00	14.540,44	64,00	17.388,32
ANEST	570,00	207.753,90	557,00	125.543,70	1.127,00	333.297,60
ANESYOĞ	67,00	611.404,22	250,00	1.754.238,36	317,00	2.365.642,58
ASTIM	366,00	29.882,09	469,00	42.000,90	835,00	71.882,99
BC	4.424,00	2.313.702,22	7.728,00	2.938.717,60	12.152,00	5.252.419,82
BCYB	40,00	345.385,28	78,00	625.148,68	118,00	970.533,96
BIYO	1.765,00	66.806,34	579,00	18.434,26	2.344,00	85.240,60
Ç.H	994,00	289.399,80	78,00	5.685,54	1.072,00	295.085,34
Ç.A	3.405,00	216.739,55	4.622,00	340.353,59	8.027,00	557.093,14
Ç.AL	730,00	67.589,48	2.914,00	295.344,23	3.644,00	362.933,71
Ç.C	1.755,00	597.688,19	2.864,00	706.538,94	4.619,00	1.304.227,13
Ç.E	1.980,00	120.514,93	3.622,00	228.258,39	5.602,00	348.773,32
Ç.H	6.533,00	1.052.925,57	5.780,00	814.736,85	12.313,00	1.867.662,42
Ç.K	539,00	30.295,94	2.156,00	198.870,22	2.695,00	229.166,16
Ç.N	1.222,00	119.664,38	2.309,00	312.325,14	3.531,00	431.989,52
Ç.P	2.144,00	108.706,56	2.284,00	122.324,49	4.428,00	231.031,05
DERM	7.481,00	414.963,33	9.480,00	492.483,40	16.961,00	907.446,73
DHL	4.873,00	337.863,36	5.868,00	423.370,83	10.741,00	761.234,19
DIA	0,00	0,00	1,00	22,10	1,00	22,10
DIS	1.896,00	71.716,89	2.157,00	81.392,78	4.053,00	153.109,67

Şekil 6.12 OLAPPROG Excel belgesi

6.5 Ayarlar

OLAPPROG kullanıcının isteklerine göre, görsel olarak özelleştirilebilir veya veri alanları, ölçütler küp görüntüsü alındıktan sonra da değiştirilebilir. Bu işlemler menüden Ayarlar kısmı seçilerek gerçekleştirilebilir. Şekil 6.13'te görsel ayarlar ekranı gösterilmiştir.



Şekil 6.13 Görsel ayarlar ekranı

Daha önce küp ayarları seçim ekranında, küp görünümü oluşturulmadan seçilebilen data görünüm özellikleri, küp görüntüsü oluşturulduktan sonra da menülerden veri alanı ayarları kısmından değiştirilebilir. Şekil 6.14'te tümünü göster seçeneği seçilerek küp görüntüsünün değiştirilebilir hale gelmesi gösterilmiştir.

HASTANE BİLGİ YÖNETİM SİSTEMİ

KARAR DESTEK RAPORLARI

- Hasta Kimlik Bilgileri
- PAKET BİLGİLERİ**
 - Paket Normal Karşılaştırma
 - Hizmet Türü Bazında Paket Normal Karşılaştırması
 - Paket Ameliyat Bilgileri
 - Hasta Durumuna göre Gelir Hesabı
 - Kesilen Fatura ve Geri Ödeme Bilgisi
- TANI BİLGİLERİ**
 - Yatan Hasta Tanı Bilgileri
 - Ayaktan Hasta Tanı Bilgileri
 - Tanı Hasta Profili
- HİZMET BİLGİLERİ**
 - Hizmet Türüne Göre İşlem Bilgileri
 - Hizmet Türüne Göre Step Değerleri
 - Hasta Türüne Göre Faturalama Bilgileri
- MAKBUZ VE KATKI PAYI BİLGİLERİ**
 - Hasta Türüne Göre Katkı Payı Bilgisi
 - Uzman Doktora Göre Katkı Payı Bilgisi
 - Hasta Makbuz Bilgileri
 - Vaka Başı Normal Karşılaştırma
- PERFORMANS BİLGİLERİ**
 - İsim Girişi Olan İşlemlerin Performans Bilgileri
 - Konsültan Unaylı İşlemlerin Performans Bilgileri

Measures

- Geliş Sayısı
- Maliyet
- Tutar
- Yatış Süresi
- Anlık Yaş
- Bölüm
- Cinsiyet
- Geliş Tarihi
- Hasta Türü
- Yatış Süresi

Table

Bölüm	Cinsiyet	Measures		Total	
		Geliş Sayısı	Maliyet	Geliş Sayısı	Maliyet
GAST	Bay	4	3.320,69	5	6.997,49
GC	Bayan	200	248.268,74	235	363.231,83
GH	Bayan	20	18.492,27	3	4.300,50
GÖGUSCER	Bayan	13	26.875,57	5	8.448,70
GÖZ	Bayan	1.372	1.333.924,71	1.528	1.410.800,94
HEMO	Bayan	4	10.157,33	6	5.755,14
KACIL	Bayan	23	70.127,17	8	21.021,98
KALP	Bayan	555	3.860.454,06	241	1.800.401,00
KALPYB	Bayan	1	13.238,44		
KARD	Bayan	1.951	3.207.859,93	958	1.405.542,42
KBB	Bayan	386	253.451,31	240	176.050,88
KDH	Bayan	5	5.620,38	1.132	1.220.640,84
MEME	Bayan			3	5.956,22
NEF	Bayan	19	14.538,44	11	15.063,81
ONKO	Bayan	3	327,14	2	212,10
ORT	Bayan	218	294.668,33	105	141.817,82
PC	Bayan	127	116.108,08	114	58.011,44
RAD	Bayan	57	30.040,02	29	10.797,72
ROM	Bayan			1	2.008,60
ÜRD	Bayan	291	445.367,40	67	85.677,31
YDYB	Bayan	2	3.092,87	1	3.292,30
YÖĞÜN	Bayan	3	6.762,60	4	32.368,10
Total		5.551	10.299.340,23	4.833	6.960.691,89

2.Paket Hastaların bölüme göre

a.Maliyet,
b.İşlem Tutarı
c.Hastaların yatış süreleri

Bütçe uygulama talimatnamesine göre bazı işlemler paket olarak değerlendirilip bu fiyattan faturalandırma işlemi yapılır. Hastalara yapılan işlemlerin toplamı ve geri ödeme kurumları tarafından ödenecek olan paket bedelleri karşılanarak paket hastaların maliyet-kazanım analizi yapılır. Hastane hizmetlerinin etkin ve verimli devam edebilmesi için paket hasta bedelleriyle gerçek maliyetlerin yöneltme belirlenmiş bir dengede tutulması gerekir. Karar noktasında paket işlemlerin hangi hasta profillerinde ne kadara mal olduğu bilgisi değişiklikle gösterildiğinden

Şekil 6.14 Veri alanı ayarları – tümünü göster

Alanları ters çevir, satır sütun alanlarının yer değiştirmesini sağlar. Düzen menüsü ise kullanıcının OLAPPROG küp işlemlerini geri almasını, yenilemesini veya OLAPGRID üzerinde istenen bir veri alanı bölgesini kopyalamasını sağlar.

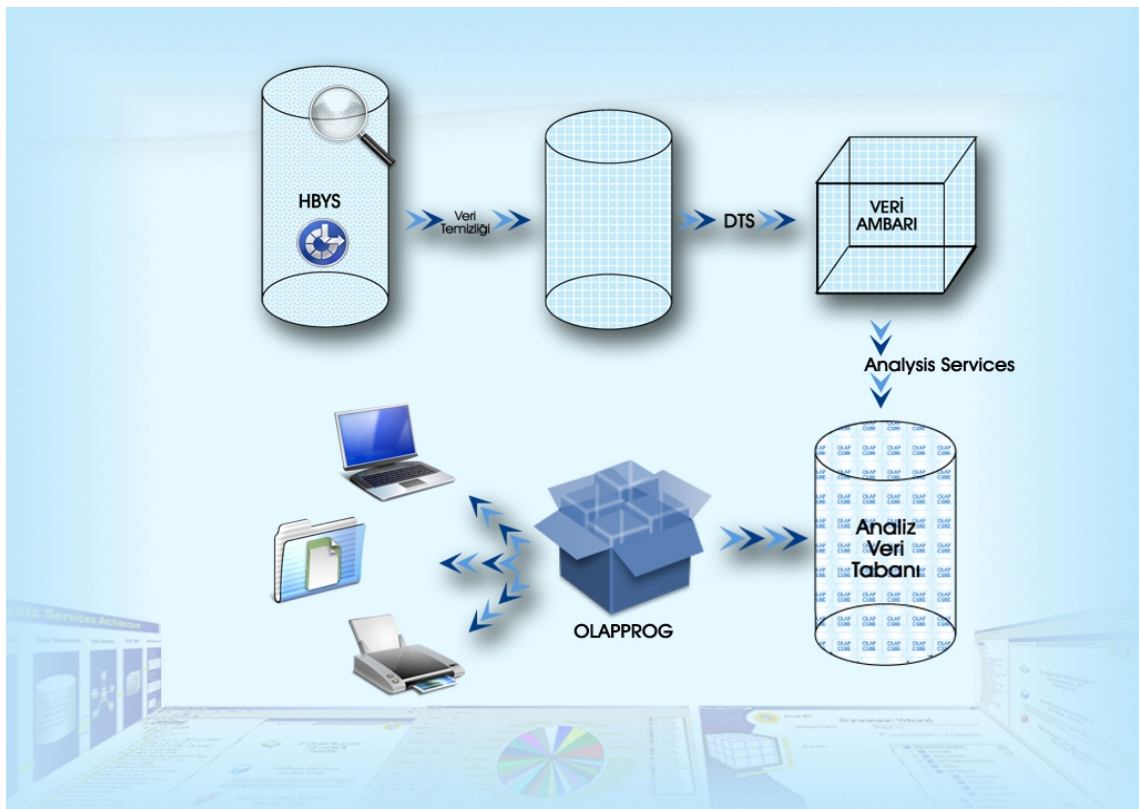
OLAPPROG'dan elde edilen diğer örnek çıktı, grafik ve küp görünümüne EK-1'de yer verilmiştir.

6.6 OLAPPROG'un Farklı HBYS'lerde Uygulanabilirliği

OLAPPROG yazılımının farklı HBYS'lerde uygulanması için izlenmesi gereken adımlar aşağıdaki gibi öngörülebilir:

1. İstenen karar destek noktalarının kararlaştırılması ve ilgili verilerin tutulduğu ilişkisel veritabanı tablolarında veri temizliği işleminin yapılması.
2. Dördüncü bölümde anlatıldığı gibi, ihtiyaç duyulan karar destek raporu sorgularının ve fact table oluşturma sorgularının belirlenmesi ve küp yapılarına karar verilmesi.

Çalışmada gerçekleştirilen diğer işlemler; verilerin DTS ile taşınması ve zamanlanması, Analysis Services kullanarak OLAP küplerinin oluşturulması ve hedef sunucu veritabanı HospitalTEMP'de küplerin kaydının yapılması adımları Microsoft SQL Server veritabanı yönetim sistemi kullanan her HBYS için ortak adımlar olup, bu işlemler tamamlandıktan sonra OLAPPROG benzeri bir uygulama geliştirilerek ya da Excel kullanılarak belirlenen karar destek noktalarındaki veri küplerine ulaşılabilir. Şekilde 6.15'te çalışmanın gerçekleştirme adımları gösterilmiştir.



Şekil 6.15 Proje geliştirme adımları

7. SONUÇLAR

Çalışmada sağlık yöneticilerine karar destek noktasında yeni bir açılım getirilerek PAÜ HBYS ile entegre OLAP tabanlı bir yazılım geliştirilmiştir. Yazılım, kullanıcısının teknik elemana ihtiyaç duymadan değişik bakış açılarına göre istediği bilgiye ulaşmasını sağlamaktadır.

Özellikle üst düzey sağlık yöneticileri, kendilerine verilen raporlardaki bakış açılarına mahkum olmadan basit mouse hareketleri ile ilginç sonuçlara ulaşabilirler. Örnek olarak Ek-1 ikinci şekilde de görülebileceği gibi, toplam paket işlem tutarlarının yaklaşık %90'ının, sadece yedi bölüm tarafından gerçekleştirilmesini verebiliriz.

Geliştirilen yazılımın, kurumsal bir bilgilendirme ağı oluşturması değil, tek kullanıcıya yönelik alternatif bilgi kaynağı olması hedeflenmiştir. Üst düzey yöneticinin orta düzeydeki yöneticilerle karar alma aşamasında etkileşim kurması ayrı bir çalışma ile değerlendirilebilir. Yazılım farklı HBYS'lere kolaylıkla adapte edilebilir ve ihtiyaç duyulduğunda yazılıma yeni karar destek noktaları için yeni OLAP küpleri de eklenebilir. Yazılım, Microsoft Analysis Services ile etkileşimli çalışarak yeni OLAP küplerini kullanıcıya sunar.

Çalışma; HBYS geliştiricileri için örnek teşkil ederek veriden bilgiye ulaşım konusunda farklı bir bakış açısı sunmuştur. Aynı zamanda hastane yöneticilerinin ve çalışanlarının, kritik karar destek noktalarında ihtiyaç duydukları bilgiye kolaylıkla ulaşmalarını sağlamıştır.

Çalışmanın bir sonraki safhası ise geçmiş verilerden yola çıkılarak ileriye yönelik verilerin tahmin edilmesi amacıyla veri madenciliği yapılmasıdır. Artificial intelligence, fuzzy logic, machine learning, neural networks gibi disiplinler ile sonuca ulaşmaya çalışılabilir. Yapılacak çalışma sonucunda örneğin bir sonraki yılın bölümlere göre hasta sayıları veya performansları hesaplanabilir. Kesilecek fatura tutarları ve geri ödemeler öngörülerek yıllık bütçe hazırlanabilir.

KAYNAKLAR

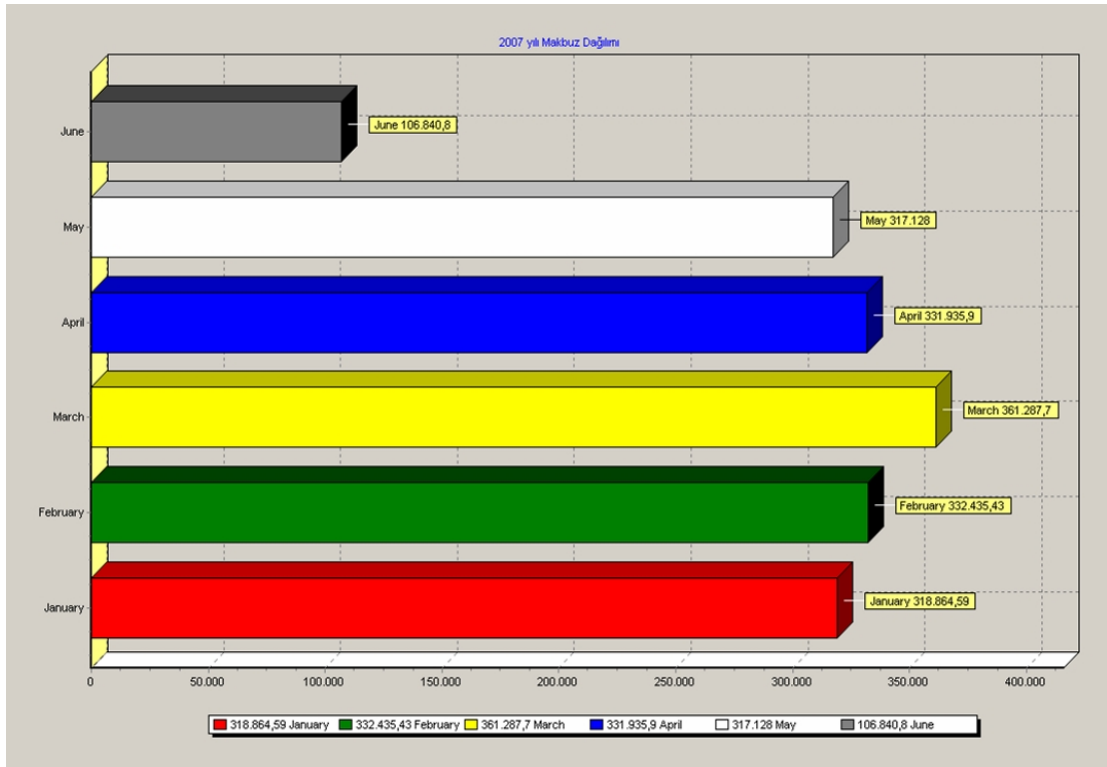
- Anonim (2004) “Hastane Bilgi Sistemleri Alımı Çerçeve İlkeleri”, T.C. Sağlık Bakanlığı Bilgi İşlem Daire Başkanlığı. s.16-18. http://sbu.saglik.gov.tr/esaglik/Documents/Hbs/hbsaci_2004.zip (25.12.2007).
- Anonim (2005) “Sağlıkta Dönüşüm Projesi Sağlık Bilgi Sistemi Kurulması Bilgi İsteme Dökümanı”, T.C. Sağlık Bakanlığı Bilgi İşlem Daire Başkanlığı. s.29-35. http://www.saglik.gov.tr/sb/usbs_bid_rev1.0.doc (25.12.2007).
- Atak, İ. (2006) Karar Destek Sistemleri (DSS), <http://www.belgeci.com/Sayfa.asp?islem=4&SayfaNo=299> (25.12.2007).
- Bain, T., Benkovich, M., Dewson, R., Ferguson, S., Graves, C., Joubert, T. J., Lee, D., Scott, M., Skoglund, R., Turley, P. and Youness, S. (2002). Professional SQL Server 2000 Data Warehousing with Analysis Services, **Wrox Press**, 700s.
- Çetinyokuş, T. ve Gökçen, H. (2002) Borsada Göstergelerle Teknik Analiz İçin Bir Karar Destek Sistemi, **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi**, http://www.mmf.gazi.edu.tr/journal/2002_1/43-58.pdf (25.12.2007).
- Inmon, W. H. (2000) Olap And Data WareHouse, http://www.datawarehouse.inf.br/Papers/Inmon_olapdw-1.pdf (23.12.2007).
- Karaca, M. S. (2006) Bilgisayar Uzmanlığı, **Nobel Yayın Dağıtım**, İstanbul, 528s.
- Koorhan, L. (2005) SQL Server 2000: OLAP Cubes and Queries Professional Skills Development, <http://www.appdev.com/documents/outlines/sql2000OLAP.pdf> (25.12.2007).
- Laudon, C. K. and Laudon, J. P., Management Information Systems, 7th Edition, **Prentice Hall**, New Jersey, 584s.
- Microsoft Corporation (2000) Course 2074 SQL Server 2000- Designing And Implementing Olap Solutions, **Microsoft Press**, 759s.
- Tiedrich, A. (2000) Microsoft Corp. SQL Server 2000 Analysis Services, http://www.gartner.com/DisplayDocument?doc_cd=93178 (24.12.2007).
- WEB_1. (2007). Karar Destek Sistemleri. <http://www.odew.net/sosyalbilimler/odewnet379.doc> (25.12.2007).
- WEB_2. (2007). Nominalia. www.lanzarotecaliente.com (25.12.2007).
- WEB_3. (2007). Interface LTD. <http://www.interface.ru/profile/index.htm> (25.12.2007).
- WEB_4. (2006). Bilişim Dünyası. <http://www.bilisimdunyasi.net.tr/yazi.asp?sayi=5&yazi=407> (25.12.2007).

WEB_5. (2007). Saęlık Bilgi Yönetim Sistemleri. <http://www.4tbilisim.com/sbys.htm> (25.12.2007).

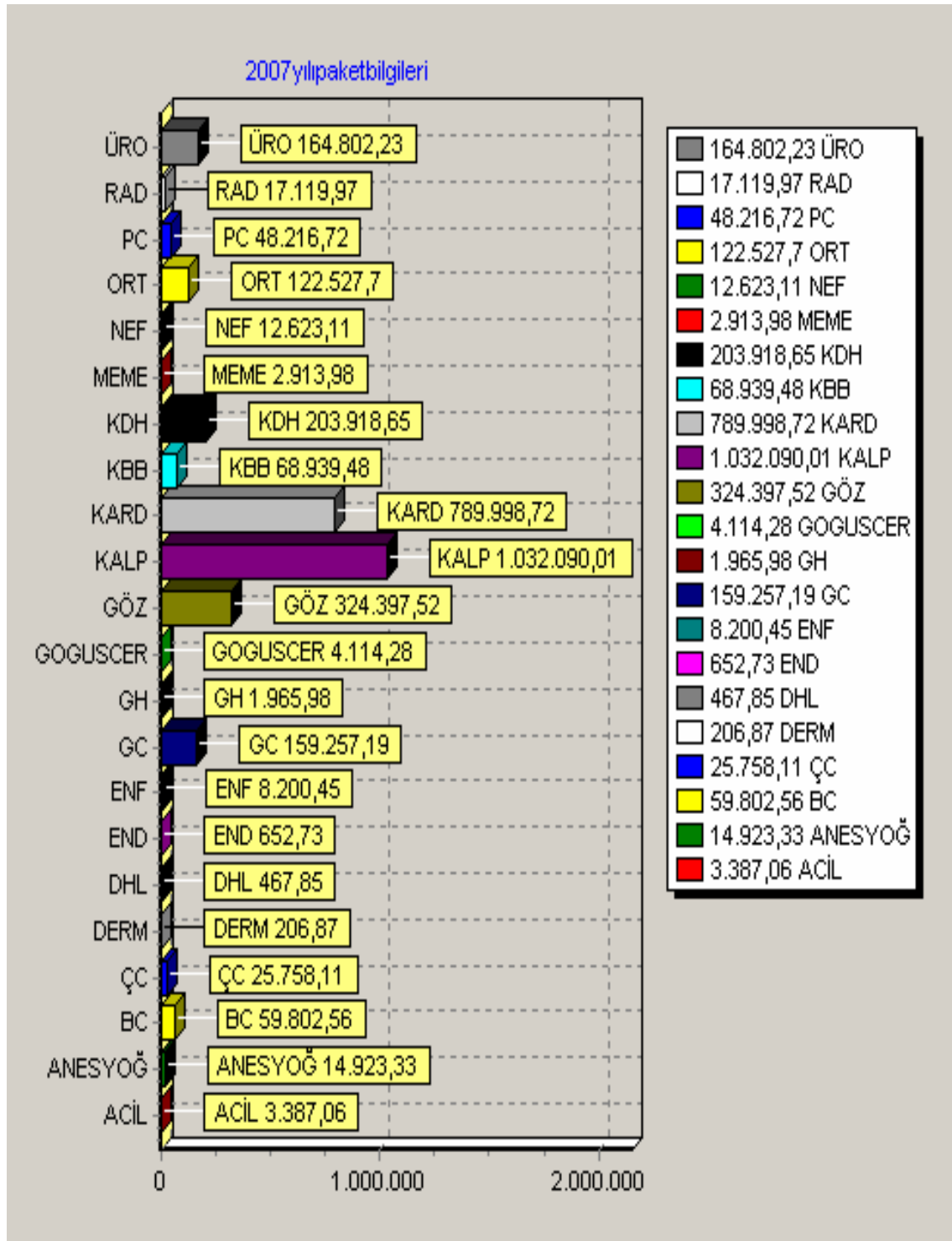
WEB_6. (2007). The OLAP Report: Architectures. <http://olapreport.com/Architectures.htm> (25.12.2007).

EKLER

Ek – 1 Program Örnek Çıktıları



2007 yılı makbuz dağılımı



2007 yılı ilk altı ayı paket bilgileri

Year	2004	2005	2006	2007	Total
Measures	Hizmet Tutarı	Hizmet Tutarı	Hizmet Tutarı	Hizmet Tutarı	Hizmet Tutarı
Poliklinik					
ACOL	21.7	6998.86	446193.76	220518.42	673732.740000002
ADT			842.39	40	882.39
A RI			15.5	15.5	31
ANEST		49.84	10887.64	3218.41	14155.89
ANESYO				3	3
ASTIM			19461.2	13087.95	32549.15
BC	73	1945.42	193192.45	76177.9799999999	271388.8499999999
BÖYO		206.73	46399.52	5055.26	51661.51
Ç.H			17876.05	2813.77	20689.82
ÇA	41.02	757.1	151918.26	45376.91	198093.29
ÇAL			27720.77	85773.3099999999	113494.08
ÇC		213.26	30263.88	17406.96	47884.1
ÇE		1384.72	77264.37	64188.1799999999	142837.27
ÇH	73	5152.42	232313.72	93542.7500000001	331081.8899999999
ÇK			19687.66	41917.4	61605.0599999999
ÇN		112.63	36241.89	31685.32	68039.8400000001
ÇP		160.12	71252.5900000002	36879.3200000001	108292.03
DERM		3180.61	205360.42	104565.22	313106.25
DHL	20	3447.37	143013.59	96043.2	242524.16
D0^		843.03	70351.91	25000.32	96195.26
DOYET			355.41		355.41
END	13	8742.75	679471.480000002	292415.22	980642.4499999999
ENF		2788.84	102430.08	51978.8899999999	157197.81
FT		4694.42	315913.35	196101.95	516709.72
GAST		3383.82	186279.67	158684.77	348348.26
GC		1358.27	146841.64	77447.4099999999	225647.32
GH	223.2	2834.09	125689.88	31000.54	159747.71
GNTK		143.56	40911.32	23490.3	64545.1799999999
GOGUSCER		1074.27	7211.29	829.04	9114.6
GÖZ	244	1336.5	763591.8599999996	242428.4900000001	1007600.8499999999
HEMO		1740.02	152817.66	58372.55	212930.23
IAH			496.02	432.18	928.2
KALP	57.7	900.65	63656.64	32084.31	96699.3
KARD	73	6515.05	404388.86	100549.59	511526.4999999997
KBB	166.8	2833.12	254211.8999999999	79406.5800000001	336618.3999999998
KDH		563.9	567432.4300000001	280808.64	848804.9699999999
KDHS			31.28		31.28
MEME		377.43	38349.3600000001	21058.6	59785.3900000002
MOKRO		4.15	1049.78	159.61	1213.54
NEF		81214.86	482460.54	117200.08	680875.4799999999
NÖR		5441.4	316423.14	109147.98	431012.52
NTIP		455.47	236031.1100000001	79003.9500000001	315490.5300000002
ONKO		1831.2	341125.8799999999	199681.97	542639.0499999999
ONP			6409.27	18344	24753.27

Year	2004	2005	2006	2007	Total
Measures	Hizmet Tutarı	Hizmet Tutarı	Hizmet Tutarı	Hizmet Tutarı	Hizmet Tutarı
Poliklinik					
ORT	86	3193.89	257174.840000001	102351.62	362806.350000001
PC		341.31	58464.61	16146.34	74952.26
PSK		367.5	366426.47	213167.28	579961.250000003
PT		855.08	71915.2299999999	11324.77	84095.08
RAD		1065.94	86263.3200000001	17968.33	105297.59
RADON			6288.14	963.13	7251.27
RAPOR		52.88	11772.9		11825.78
ROM	13	2787.38	141139.98	44734.25	188674.61
TONKO			769.21	427.23	1196.44
UYKU			96	241.5	337.5
ÜRO	31.7	2386.06	194936.26	61265.32	258619.34
YDYB			1699.18	4026.77	5725.95
Total	1137.12	163735.92	8230783.55999986	3606552.37000001	12002208.97

Bölmelere ve yıllara göre hizmet tutarı

Year	2006ONCESI		2006SONRASI		Total	
Measures	Düşüm Tutarı		Düşüm Tutarı		Düşüm Tutarı	
Hasta Türü	Value	Percent of Row Total	Value	Percent of Row Total	Value	Percent of Row Total
	1683,81	1683,81	18,39	18,39	1702,2	1702,2
Bağkurlu	276937,67	276937,7	759486,9	759486,9	1036424,57	1036425
Emekli	6430779,1	6430779	7128789,79	7128790	13559568,9	13559569
Hukuk			51,81	51,81	51,81	51,81
Sevklı	5930038,9	5930039	7022276,4	7022276	12952315,3	12952315
Sosyal	28738,81	28738,81	23660,61	23660,61	52399,42	52399,42
SSK	5132636,9	5132637	11134518,5	11134518	16267155,4	16267155
SSKPAKE	96214,14	96214,14	2393,45	2393,45	98607,59	98607,59
Ücretli	774611,25	774611,3	1016343,17	1016343	1790954,42	1790954
Yeşil Kart	2003376	2003376	2839356,62	2839357	4842732,59	4842733
Total	20675017	20675017	29926895,6	29926896	50601912,2	50601912

Yıllara göre geri ödeme bilgisi

Year	2002			2003		
Measures	Geliş Sayısı	Maliyet	Tutar	Geliş Sayısı	Maliyet	Tutar
Hasta Türü						
Bağkurlu	1	1148,23	740	97	231556,55	153410,9
Emekli	270	583008,53	359454,05	477	956881,15	554986,37
Personel						
Sevklı				1	1194,53	625
Sorunlu						
Sosyal						
SSK				486	1072935,19	742770,89
SSKPAKET				1	1586,48	582,7
Ücretli				34	99678,79	70526,65
Yeşil Kart						
Total	271	584156,76	360194,05	1096	2363832,69	1522902,51

2004			2005		
Geliş Sayısı	Maliyet	Tutar	Geliş Sayısı	Maliyet	Tutar
202	530769,29	336600,24	12	30812,09	29366,56
442	790619,68	493165,28	411	405186,93	346603,99
			1	272,27	461,44
6	9339,37	5882,9	408	336086,63	292415,49
1	2149,24	1500			
			2	1283,06	1384,32
828	1779675,98	926682,75	652	660240,97	556781,22
58	56642,41	41244,7	14	11256	9605,19
67	152340,62	79522,35	34	33289,3	28058,78
1	961,57	625	100	119694,34	104513,93
1605	3322498,16	1885223,22	1634	1598121,59	1369190,92

2006			2007	
Geliş Sayısı	Maliyet	Tutar	Geliş Sayısı	Maliyet
314	909198,54	696019,87	155	453086,39
977	1335942,06	1086355,83	466	596818,33
898	989343,28	815343,77	395	398178,69
8	8323,47	5302,63	2	1498,84
1324	2194151,57	1819541,15	685	1215896,05
98	224580,53	173952,19	31	71941,8
292	663598,97	590372,97	133	328864,4
3911	6325138,42	5186888,41	1867	3066284,5

Total			
Tutar	Geliş Sayısı	Maliyet	Tutar
386560,37	781	2156571,09	1602697,94
518404,46	3043	4668456,68	3358969,98
	1	272,27	461,44
342495,08	1708	1734142,5	1456762,24
	1	2149,24	1500
1108,14	12	11105,37	7795,09
1103126,95	3975	6922899,76	5148902,96
	73	69484,89	51432,59
65525,21	264	581831,04	417585,18
354662,67	526	1113119,28	1050174,57
2771882,88	10384	17260032,12	13096281,99

Yıllara göre paket maliyet kazanım değerleri

Ek – 2 Veri Temizleme SQL İfadeleri

```
Delete th_dosya where dosyano not in (SELECT distinct
D.dosyano
```

```
FROM TH_Dosya D inner join th_refkart R on D.dosyano =
R.dosyano
```

```
WHERE (Cinsiyet IS NULL)) and Cinsiyet is null
```

```
Update Th_dosya set Cinsiyet = null where Cinsiyet = ''
```

```
Update Th_dosya set Cinsiyet = 'Bay' where Cinsiyet = 'Erkek'
```

```
Update Th_dosya set Cinsiyet = 'Bayan' where Cinsiyet = 'Kız'
```

```
update TH_Dosya set Cinsiyet = 'Bay' WHERE (Yakinlik = 'OĞLU'
or Yakinlik = 'BABASI') and Cinsiyet is null
```

```
update TH_Dosya set Cinsiyet = 'Bayan' WHERE (Yakinlik =
'KIZI' or Yakinlik = 'ANNESİ') and Cinsiyet is null
```

```
Update TH_Dosya set Cinsiyet = 'Bayan' WHERE (Cinsiyet IS
NULL) and Yakinlik = 'EŞİ'
```

```
SELECT * FROM TH_Dosya WHERE (Cinsiyet IS NULL) AND (Yakinlik
IN ('EŞİ','OKIZI', 'KZI' , 'OGLU'))
```

```
SELECT * FROM TH_Dosya where isdate(dogtarih)<>1 and
(dogtarih IS NOT NULL)
```

```
update th_dosya set dogtarih = replace(dogtarih,',','.'))
```

```
SELECT * FROM TH_Dosya WHERE (ISDATE(DogTarih) <> 1) AND
(DogTarih IS NOT NULL) AND (DogTarih <> '')
```

```
update th_dosya set dogtarih = null where (DogTarih = '')
```

```
SELECT * FROM TH_Dosya WHERE (ISDATE(DogTarih) <> 1) AND
(DogTarih IS NOT NULL) AND (DogTarih <> '') AND ({ fn
LENGTH(DogTarih) } <> 4)
```

```
update th_dosya set dogtarih = null where { fn
LENGTH(DogTarih) } = 1
```

```
SELECT * FROM TH_Dosya WHERE (ISDATE(DogTarih) <> 1) AND
(DogTarih IS NOT NULL) AND ({ fn LENGTH(DogTarih) } = 4) AND
(LEFT(DogTarih, 2) <> '13')
```

```
UPDATE TH_Dosya SET DogYer = '2002' WHERE (DogYer = 'AKALAN'
OR DogYer = 'ALAATTİN' OR DogYer = 'DARIVEREN' OR DogYer =
'DEDEBAĞI' OR DogYer = 'DODURGALAR' OR DogYer = 'GÖLCÜK' OR
DogYer = 'KELEKÇİ' OR DogYer = 'KUMAFŞARI' OR DogYer =
'YASSIHÜYÜK' OR DogYer = 'YAZIR' OR DogYer = 'YEŞİLDERE' OR
DogYer = 'YEŞİLYUVA' OR DogYer = 'YUMRUTAŞ' OR Dogyer = 'DEDEBAĞ
```

```
' ) UPDATE TH_Dosya SET DogTarih = DogTarih+584 WHERE
ISDATE(DogTarih) <> 1) AND (DogTarih IS NOT NULL)
```

```
delete th_dosya where dosyano not in (SELECT distinct
D.dosyano
```

```
FROM TH_Dosya D inner join th_refkart R on D.dosyano =
R.dosyano
```

```
WHERE (Cinsiyet IS NULL)) and Cinsiyet is null
```

```
Update Th_dosya set Cinsiyet = null where Cinsiyet = ''
```

```
Update Th_dosya set Cinsiyet = 'Bay' where Cinsiyet = 'Erkek'
```

```
Update Th_dosya set Cinsiyet = 'Bayan' where Cinsiyet = 'Kız'
```

```
update TH_Dosya set Cinsiyet = 'Bay' WHERE (Yakinlik = 'OĞLU'
or Yakinlik = 'BABASI') and Cinsiyet is null
```

```
update TH_Dosya set Cinsiyet = 'Bayan' WHERE (Yakinlik =
'KIZI' or Yakinlik = 'ANNESİ') and Cinsiyet is null
```

```
Update TH_Dosya set Cinsiyet = 'Bayan' WHERE (Cinsiyet IS
NULL) and Yakinlik = 'EŞİ'
```

```
SELECT * FROM TH_Dosya WHERE (Cinsiyet IS NULL) AND (Yakinlik
IN ('EŞİ', 'OKIZI', 'KZI', 'OGLU'))
```

```
Update TH_Dosya Set Cinsiyet = 'Bay' WHERE (Cinsiyet IS NULL)
and
```

```
(adsoy like 'murat%' or adsoy like 'ahmet%' or adsoy like
'ali%' or adsoy like 'mehmet%' or adsoy like 'uğur%' or adsoy
like 'hasan%' or adsoy like 'hüseyin%' or adsoy like 'muhammet%'
or adsoy like 'abdullah%' or adsoy like 'adem%' or adsoy like
'bayram%' or adsoy like 'bülent%' or adsoy like 'abdullah%' or
adsoy like 'adem%' or adsoy like 'bayram%' or adsoy like
'durmuş%' or adsoy like 'fatih%' or adsoy like 'gökhan%' or
adsoy like 'ibrahim%' or adsoy like 'hakan%' or adsoy like
'halil%' or adsoy like 'ismail%' or adsoy like 'kadir%' or adsoy
like 'kemal%' or adsoy like 'levent%' or adsoy like 'M.%' or
adsoy like 'mahmut%' or adsoy like 'mesut%' or adsoy like
'mustafa%' or adsoy like 'osman%' or adsoy like 'orhan%' or
adsoy like 'ömer%' or adsoy like 'özcan%' or adsoy like
'ramazan%' or adsoy like 'raşit%' or adsoy like 'recep%' or
adsoy like 'sabri%' or adsoy like 'serkan%' or adsoy like
'süleyman%' or adsoy like 'şakir%' or adsoy like 'şükrü%' or
adsoy like 'tahsin%' or adsoy like 'veli%' or adsoy like
'veysel%' or adsoy like 'yaşar%' or adsoy like 'yılmaz%' or
adsoy like 'yunus%' or adsoy like 'yusuf%' )
```

```
Update TH_Dosya Set Cinsiyet = 'Bayan' WHERE (Cinsiyet IS
NULL) and (adsoy like 'zeynep%' or adsoy like 'zehra%' or adsoy
like 'yıldız%' or adsoy like 'ümmü%' or adsoy like 'tuğ%' or
```

adsoy like 'şerife%' or adsoy like 'sultan%' or adsoy like 'sibel%' or

adsoy like 'n%' or adsoy like 'mer%' or adsoy like 'mel%' or adsoy like 'kez%' or adsoy like 'ilk%' or adsoy like 'hülya%' or adsoy like 'hu%' or adsoy like 'ha%' or adsoy like 'gül%' or adsoy like 'fat%' or adsoy like 'emine%' or adsoy like 'elif%' or adsoy like 'ebru%' or adsoy like 'dudu%' or adsoy like 'dilek%' or adsoy like 'cennet%' or adsoy like 'cemile%' or adsoy like 'ay%' or adsoy like 'emel%' or adsoy like 'mü%')

```
CREATE VIEW dbo.cinsiyet AS
```

```
SELECT R.*, D.Cinsiyet AS cinsiyet FROM dbo.TH_Dosya D INNER JOIN
```

```
dbo.TH_RefKart R ON D.DosyaNo = R.DosyaNo WHERE (D.Cinsiyet IS NULL)
```

```
Update cinsiyet set cinsiyet = 'Bayan' WHERE (Yakinlik in ('EŞİ','ANNESİ','KIZI'))
```

```
Update cinsiyet set cinsiyet = 'Bay' WHERE Yakınlık = 'OĞLU'
```

```
Update cinsiyet set cinsiyet = 'Bayan' WHERE Polkod = 'KDH'
```

```
select * from TH_Dosya where DogYer = 'DENİZLİ'
```

```
UPDATE TH_Dosya SET DogYer = '2001' WHERE (DogYer = 'DENİZLİ')
```

```
UPDATE TH_Dosya SET DogYer = '2001' WHERE (DogYer like 'dnz%')
```

```
select * from th_Dosya where dogyer like 'AC%' and dogyer not in ('ACIGÖL','ACILAR','ACARLAR')
```

```
UPDATE TH_Dosya SET DogYer = '2002' Where dogyer like 'AC%' and dogyer not in ('ACIGÖL','ACILAR','ACARLAR')
```

```
select * from th_Dosya where dogyer like 'AKKÖY'
```

```
UPDATE TH_Dosya SET DogYer = '2003' WHERE (DogYer = 'AKKÖY')
```

```
select * from th_Dosya where dogyer like 'BABADAĞ'
```

```
UPDATE TH_Dosya SET DogYer = '2004' WHERE (DogYer = 'BABADAĞ')
```

```
select * from th_Dosya where dogyer like 'BAKLAN'
```

```
UPDATE TH_Dosya SET DogYer = '2005' WHERE (DogYer = 'BAKLAN')
```

```
select * from th_Dosya where dogyer like 'BEKİLLİ'
```

```
UPDATE TH_Dosya SET DogYer = '2006' WHERE (DogYer =
'BEKİLLİ')

select * from th_Dosya where dogyer like 'BEYAĞAÇ'

UPDATE TH_Dosya SET DogYer = '2007' WHERE (DogYer =
'BEYAĞAÇ')

select * from th_Dosya where dogyer like 'BOZKURT'

UPDATE TH_Dosya SET DogYer = '2008' WHERE (DogYer =
'BOZKURT')

select * from th_Dosya where dogyer like 'BULDAN'

UPDATE TH_Dosya SET DogYer = '2009' WHERE (DogYer = 'BULDAN')

select * from th_Dosya where dogyer like 'ÇAL'

UPDATE TH_Dosya SET DogYer = '2010' WHERE (DogYer = 'ÇAL')

select * from th_Dosya where dogyer like 'ÇAMELİ'

UPDATE TH_Dosya SET DogYer = '2011' WHERE (DogYer = 'ÇAMELİ')

select * from th_Dosya where dogyer like 'ÇARDAK'

UPDATE TH_Dosya SET DogYer = '2012' WHERE (DogYer = 'ÇARDAK')

select * from th_Dosya where dogyer like 'ÇİVRİL'

UPDATE TH_Dosya SET DogYer = '2013' WHERE (DogYer = 'ÇİVRİL')

select * from th_Dosya where dogyer like 'GÜNEY'

UPDATE TH_Dosya SET DogYer = '2014' WHERE (DogYer = 'GÜNEY')

select * from th_Dosya where dogyer like 'HONAZ'

UPDATE TH_Dosya SET DogYer = '2015' WHERE (DogYer = 'HONAZ')

select * from th_Dosya where dogyer like 'KALE'

UPDATE TH_Dosya SET DogYer = '2016' WHERE (DogYer = 'KALE')

select * from th_Dosya where dogyer like 'SARAYKÖY'

UPDATE TH_Dosya SET DogYer = '2017' WHERE (DogYer =
'SARAYKÖY')

select * from th_Dosya where dogyer like 'SERİNHİSAR'

UPDATE TH_Dosya SET DogYer = '2018' WHERE (DogYer =
'SERİNHİSAR')

select * from th_Dosya where dogyer like 'TAVAS'
```

```

UPDATE TH_Dosya SET DogYer = '2019' WHERE (DogYer = 'TAVAS')

SELECT * FROM TH_Dosya WHERE (LEN(DogYer) < 3) ORDER BY
DogYer

Select Dogyer, count(*) as Sayi from Th_Dosya Where
isnumeric(Dogyer) = 0 Group by Dogyer order by Sayi desc

select * from th_Dosya where dogyer like 'DİNAR'

UPDATE TH_Dosya SET DogYer = '0307' WHERE (DogYer = 'DİNAR')

select * from th_Dosya where dogyer like 'İZMİR'

UPDATE TH_Dosya SET DogYer = '3501' WHERE (DogYer = 'İZMİR')

select * from th_Dosya where dogyer like 'ANKARA'

UPDATE TH_Dosya SET DogYer = '0601' WHERE (DogYer = 'ANKARA')

select * from th_Dosya where dogyer like 'UŞAK'

UPDATE TH_Dosya SET DogYer = '6401' WHERE (DogYer = 'UŞAK')

select * from th_Dosya where dogyer like 'AFYON'

UPDATE TH_Dosya SET DogYer = '0301' WHERE (DogYer = 'AFYON')

select * from th_Dosya where dogyer like 'BURDUR'

UPDATE TH_Dosya SET DogYer = '1501' WHERE (DogYer = 'BURDUR')

select * from th_Dosya where dogyer like 'AYDIN'

UPDATE TH_Dosya SET DogYer = '0901' WHERE (DogYer = 'AYDIN')

select * from th_Dosya where dogyer like 'İSTANBUL'

UPDATE TH_Dosya SET DogYer = '3401' WHERE (DogYer =
'İSTANBUL')

select * from th_Dosya where dogyer like 'ISPARTA'

UPDATE TH_Dosya SET DogYer = '3201' WHERE (DogYer =
'ISPARTA')

select * from th_Dosya where dogyer like 'AĞRI'

UPDATE TH_Dosya SET DogYer = '0401' WHERE (DogYer = 'AĞRI')

select * from th_Dosya where dogyer like 'KONYA'

UPDATE TH_Dosya SET DogYer = '4201' WHERE (DogYer = 'KONYA')

select * from th_Dosya where dogyer like 'ERZURUM'

```



```

UPDATE TH_Dosya SET DogYer = '2501' WHERE (DogYer =
'ERZURUM')

select * from th_Dosya where dogyer like 'MUĞLA'

UPDATE TH_Dosya SET DogYer = '4801' WHERE (DogYer = 'MUĞLA')

select * from th_Dosya where dogyer like 'DAZKIRI'

UPDATE TH_Dosya SET DogYer = '0306' WHERE (DogYer =
'DAZKIRI')

select * from th_Dosya where dogyer like 'SANDIKLI'

UPDATE TH_Dosya SET DogYer = '0311' WHERE (DogYer =
'SANDIKLI')

select * from th_Dosya where dogyer like 'NAZİLLİ'

UPDATE TH_Dosya SET DogYer = '0911' WHERE (DogYer =
'NAZİLLİ')

select * from th_Dosya where dogyer like 'GÖLHİSAR'

UPDATE TH_Dosya SET DogYer = '1504' WHERE (DogYer =
'GÖLHİSAR')

select * from th_Dosya where dogyer like 'YEŞİLOVA'

UPDATE TH_Dosya SET DogYer = '1508' WHERE (DogYer =
'YEŞİLOVA')

UPDATE Th_Dosya set Dogyer = null where Dogyer = ''

UPDATE TH_Dosya SET DogYer = '9049' WHERE (Dogyer =
'ALMANYA')

UPDATE TH_Dosya SET DogYer = null WHERE (Dogyer = 'TÜRKİYE')

CREATE VIEW dbo.Dogyer AS

Select Dogyer,İlad,İlçeAd,İlİlçeKod from Th_Dosya D inner
join Th_Kurumİlilçe İ on (D.Dogyer = İ.İlad or D.Dogyer =
İ.İlçead)

Select * from Dogyer where Dogyer = İlad

UPDATE Dogyer SET DogYer = Left(İlİlçeKod,2)+'01' where
Dogyer = İlad

UPDATE Dogyer SET DogYer = İlİlçekod where Dogyer = İlçead

UPDATE TH_Dosya SET DogYer = '2001' WHERE (DogYer =
'DENZİLİ')

```

```
UPDATE TH_Dosya SET DogYer = '2019' WHERE (DogYer = 'TAVAS
DENİZLİ')
```

```
UPDATE TH_Dosya SET DogYer = '2019' WHERE (DogYer =
'KIZILCABÖLÜK')
```

```
UPDATE TH_Dosya SET DogYer = '2009' WHERE (DogYer =
'BULDAN/DENİZLİ')
```

```
UPDATE TH_Dosya SET DogYer = '2013' WHERE (DogYer =
'ÇİVRİL/DENİZLİ')
```

```
UPDATE TH_Dosya SET DogYer = '2016' WHERE (DogYer =
'KALE/DENİZLİ')
```

```
UPDATE TH_Dosya SET DogYer = '2001' WHERE (DogYer =
'DENİZLER')
```

```
UPDATE TH_Dosya SET DogYer = '2002' WHERE (DogYer =
'A.PAYAM')
```

```
UPDATE TH_Dosya SET DogYer = '2010' WHERE (DogYer = 'ÇAL
DENİZLİ')
```

```
UPDATE TH_Dosya SET DogYer = '4101' WHERE (DogYer = 'İZMİT')
```

```
UPDATE TH_Dosya SET DogYer = '5401' WHERE (DogYer =
'ADAPAZARI')
```

```
UPDATE TH_Dosya SET DogYer = '0307' WHERE (DogYer =
'DİNAR/AFYON')
```

```
UPDATE TH_Dosya SET DogYer = '0311' WHERE (DogYer =
'SANDIKLI/AFYON')
```

```
UPDATE TH_Dosya SET DogYer = '3301' WHERE (DogYer = 'MERSİN')
```

```
UPDATE TH_Dosya SET DogYer = '3101' WHERE (DogYer =
'ANTAKYA')
```

```
UPDATE TH_Dosya SET DogYer = '2002' WHERE (DogYer = 'AKALAN'
OR DogYer = 'ALAATTİN' OR DogYer = 'DARIVEREN' OR DogYer =
'DEDEBAĞI' OR DogYer = 'DODURGALAR' OR DogYer = 'GÖLCÜK' OR
DogYer = 'KELEKÇİ' OR DogYer = 'KUMAFŞARI' OR DogYer =
'YASSIHÜYÜK' OR DogYer = 'YAZIR' OR DogYer = 'YEŞİLDERE' OR
DogYer = 'YEŞİLYUVA' OR DogYer = 'YUMRUTAŞ' OR DogYer = 'DEDEBAĞ
')
```

```
UPDATE TH_Dosya SET DogYer = '90359' WHERE DOGYER =
'BULGARİSTAN'
```

```
UPDATE TH_Dosya SET DogYer = '9031' WHERE DOGYER = 'HOLLANDA'
```

```
UPDATE TH_Dosya SET DogYer = '9033' WHERE DOGYER = 'FRANSA'
```

```
UPDATE TH_Dosya SET DogYer = '9041' WHERE DOGYER = 'İSVİÇRE'
```

```
UPDATE TH_Dosya SET DogYer = '9043' WHERE DOGYER = 'AVUSTURYA'
```

```
UPDATE TH_Dosya SET DogYer = '1508' WHERE DOGYER = 'SALDA' OR DOGYER = 'YEŞİLOVA/BURDUR'
```

```
UPDATE TH_Dosya SET DogYer = '6301' WHERE DOGYER = 'URFA'
```

```
UPDATE TH_Dosya SET DogYer = '2018' WHERE DOGYER = 'YÜREĞİL'
```

```
UPDATE TH_Dosya SET DogYer = '2010' WHERE DOGYER IN ('SÜLLER', 'İSABEY', 'HANÇALAR')
```

```
UPDATE TH_Dosya SET DogYer = '4308' WHERE DOGYER = 'DEREKÖY'
```

```
UPDATE TH_Dosya SET DogYer = '0908' WHERE DOGYER = 'DEDEKÖY'
```

```
UPDATE TH_Dosya SET DogYer = '0308' WHERE DOGYER = 'EMİRDAĞ'
```

```
UPDATE TH_Dosya SET DogYer = '0302' WHERE DOGYER = 'BAŞMAKCI'
```

```
UPDATE TH_Dosya SET DogYer = '2012' WHERE DOGYER = 'BEYLERLİ'
```

```
UPDATE TH_Dosya SET DogYer = '1507' WHERE DOGYER = 'BEYKÖY'
```

```
UPDATE TH_Dosya SET DogYer = '2008' WHERE DOGYER = 'İNCELER'
```

```
UPDATE TH_Dosya SET DogYer = '2018' WHERE DOGYER = 'KOCAPINAR'
```

```
UPDATE TH_Dosya SET DogYer = '1507' WHERE DOGYER = 'TEFENLİ'
```

```
UPDATE TH_Dosya SET DogYer = '2008' WHERE DOGYER = 'TUTLUCA'
```

```
UPDATE TH_Dosya SET DogYer = '2010' WHERE DOGYER like 'DEN%çal%' or Dogyer like '%cal'
```

```
UPDATE TH_Dosya SET DogYer = '2002' WHERE DOGYER like 'DEN%acı%'
```

```
UPDATE TH_Dosya SET DogYer = '2019' WHERE DOGYER like 'DEN%tav%'
```

```
UPDATE TH_Dosya SET DogYer = '2016' WHERE DOGYER like 'DEN%kale%'
```

```
UPDATE TH_Dosya SET DogYer = '2013' WHERE DOGYER like 'DEN%çiv%'
```

```
UPDATE TH_Dosya SET DogYer = '2012' WHERE DOGYER like 'DEN%çar%'
```

```
UPDATE TH_Dosya SET DogYer = '2018' WHERE DOGYER like 'DEN%ser%'
```

```
UPDATE TH_Dosya SET DogYer = '2009' WHERE DOGYER like 'DEN%bul%'
```

```
UPDATE TH_Dosya SET DogYer = '2017' WHERE DOGYER like 'DEN%sara%'
```

```
UPDATE TH_Dosya SET DogYer = '2015' WHERE DOGYER like 'DEN%hon%'
```

```
UPDATE TH_Dosya SET DogYer = '2001' WHERE DOGYER like 'DEN%'
```

```
UPDATE TH_Dosya SET DogYer = '2010' WHERE Dogyer like '%çal%deni%'
```

```
UPDATE TH_Dosya SET DogYer = '2011' WHERE Dogyer like '%çam%deni%'
```

```
UPDATE TH_Dosya SET DogYer = '2016' WHERE Dogyer like '%kale%deni%'
```

```
UPDATE TH_Dosya SET DogYer = '2013' WHERE Dogyer like '%çiv%deni%'
```

```
UPDATE TH_Dosya SET DogYer = '2017' WHERE Dogyer like '%sara%deni%'
```

```
UPDATE TH_Dosya SET DogYer = '2019' WHERE Dogyer like '%tav%deni%'
```

```
UPDATE TH_Dosya SET DogYer = '2018' WHERE Dogyer like '%sar%deni%'
```

```
UPDATE TH_Dosya SET DogYer = 2014 WHERE Dogyer like '%güney%'
```

```
UPDATE TH_Dosya SET DogYer = 1504 WHERE Dogyer like '%göl%'
```

```
Update th_dosya set yakinlik = 'EŞİ' where yakinlik like 'eş%'
```

```
Update th_dosya set yakinlik = 'ANNESİ' where yakinlik like 'ann%'
```

```
Update th_dosya set yakinlik = 'KENDİSİ' where yakinlik like 'kendi%'
```

```
Update th_dosya set yakinlik = 'BABASI' where yakinlik like 'bab%'
```

```
SELECT * FROM TH_Dosya WHERE (Yakinlik LIKE 'E%') AND (Yakinlik <> 'EŞİ')
```

```
SELECT Yakınlık AS yakinlik, * FROM TH_Dosya WHERE (Yakinlik IS NOT NULL) AND (Yakinlik NOT IN ( 'ANNESİ', 'BABASI', 'EŞİ', 'OĞLU', 'KIZI', 'KENDİSİ' )) ORDER BY Yakınlık asc/desc
```

```
Update th_dosya set yakinlik = null where yakinlik = ''
```

```
Update th_dosya set yakinlik = null where yakinlik like 'ARAŞ%'
```

```
UPDATE TH_Dosya SET Yakinlik = 'OĞLU'
```

```
WHERE (Yakinlik LIKE 'BE%' OR Yakinlik LIKE 'C%' OR Yakinlik LIKE 'Ç%') AND (Cinsiyet = 'Bay')
```

```
UPDATE TH_Dosya SET Yakinlik = 'KIZI'
```

```
WHERE (Yakinlik LIKE 'BE%' OR Yakinlik LIKE 'C%' OR Yakinlik LIKE 'Ç%') AND (Cinsiyet = 'Bayan')
```

```
SELECT * FROM TH_Dosya WHERE ISDATE(dogtarih)<>1 and (DogTarih IS NOT NULL)
```

```
update th_dosya set dogtarih = replace(dogtarih,',','.'))
```

```
SELECT * FROM TH_Dosya WHERE (ISDATE(DogTarih) <> 1) AND (DogTarih IS NOT NULL) AND (DogTarih <> '')
```

```
update th_dosya set dogtarih = null where (DogTarih = '')
```

```
SELECT * FROM TH_Dosya WHERE (ISDATE(DogTarih) <> 1) AND (DogTarih IS NOT NULL) AND (DogTarih <> '') AND ({ fn LENGTH(DogTarih) } <> 4)
```

```
update th_dosya set dogtarih = null where LENGTH(DogTarih) = 1
```

```
SELECT * FROM TH_Dosya WHERE (ISDATE(DogTarih) <> 1) AND (DogTarih IS NOT NULL) AND ({ fn LENGTH(DogTarih) } = 4) AND (LEFT(DogTarih, 2) <> '13')
```

```
UPDATE TH_Dosya SET DogTarih = DogTarih+584 WHERE (ISDATE(DogTarih) <> 1) AND (DogTarih IS NOT NULL)
```

```
SELECT * FROM TH_Dosya WHERE (DogTarih IS NOT NULL) AND ({ fn LENGTH(DogTarih) } = 4)
```

```
UPDATE TH_Dosya SET DogTarih = '01.01.'+DogTarih
```

```
WHERE (DogTarih IS NOT NULL) AND ({ fn LENGTH(DogTarih)} = 4)
```

```
SELECT * FROM TH_Dosya WHERE (LEN(DogTarih) <> 10)
```

ÖZGEÇMİŞ

Murat TAŞER, 1978 yılında İzmit'te doğdu. İlk, orta ve lise öğrenimi bu şehirde tamamladı. 1995 yılında başladığı Sakarya Üniversitesi Makine Mühendisliği Bölümü'nü yarıda bırakarak, bir yıl sonra girdiği Mersin Üniversitesi Bilgisayar Mühendisliği Bölümü'nü 2000 yılında bitirmiştir. Aynı yıl temmuz ayı içerisinde Pamukkale Üniversitesi Eğitim Uygulama ve Araştırma Hastaneleri Bilgi İşlem Birimi'nde çalışmaya başlamıştır. Askerliğini Aralık 2001-Nisan 2003 tarihleri arasında Erzurum Mareşal Çakmak Hastanesi Bilgi İşlem Birimi'nde OBİ subayı olarak tamamladıktan sonra tekrar Pamukkale Üniversitesi Hastaneleri'ndeki görevine Bilgi İşlem Sorumlusu olarak dönen Murat TAŞER evlidir.