

T.C.
BEYKENT UNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK BİLGİSAYAR ANABİLİM DALI
BİLGİ TEKNOLOJİLERİ BİLİM DALI

**DİNAMİK VERİ AMBARI TASARIMI VE BİR
UYGULAMA ÖRNEĞİ**
Yüksek Lisans Tezi

Tezi Hazırlayan: **Aytaç Bora KUTLU**

İSTANBUL, 2016

T.C.
BEYKENT UNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK BİLGİSAYAR ANABİLİM DALI
BİLGİ TEKNOLOJİLERİ BİLİM DALI

**DİNAMİK VERİ AMBARI TASARIMI VE BİR
UYGULAMA ÖRNEĞİ**
Yüksek Lisans Tezi

Tezi Hazırlayan: **Aytaç Bora KUTLU**
Öğrenci No: 080862006

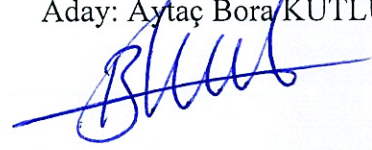
Danışman:
Doç. Dr. Gökhan SİLAHTAROĞLU

İSTANBUL, 2016

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum “Dinamik Veri Ambarı Tasarımı ve Bir Uygulama Örneği” başlıklı çalışmanın bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 15/04/2016

Aday: Aytaç Bora KUTLU



T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi 080862006 no'lu AYTAÇ BORA KUTLU'nun 16/5/16 tarihinde yapılan tez savunma sınavı¹ sonucunda 45 dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle, KABUL kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : BİLGİ TEKNOLOJİLERİ
Programı : BİLGİ TEKNOLOJİLERİ
Tez Başlığı³ :

<u>Tez Sınav Jürisi</u>	<u>Öğretim Üyesi</u>
Danışman	: DOÇ.DR. GÖKHAN SİLAHTAROĞLU
Üye	: YRD.DOÇ.DR. EDİZ ŞAYKOL
Üye	: YRD.DOÇ.DR. TURHAN KARAGÜLER

İmza



¹ Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

²Tez sınavının tamamlanmasından sonra jüri, tez hakkında “kabul”, “düzeltme” veya “red” kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişkisi kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişkisi kesilir.(Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

³İleridedoğabilecek saklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

DİNAMİK VERİ AMBARI TASARIMI VE BİR UYGULAMA ÖRNEĞİ

Tezi Hazırlayan: Aytaç Bora KUTLU

ÖZET

Bilgi kurumların sahip olduğu önemli varlıklardan birisidir ve kurumların entelektüel sermayesini oluşturur. Dijital ortamda üretilen ve saklanan veri, bilginin üretilmesini katkıda bulunur. Teknolojinin geldiği nokta ve gelişim hızı değerlendirildiğinde dijital ortamda bulunan verinin çok büyük boyutlara ulaştığı ve hızlanarak artmaya devam edeceği düşünülmektedir. Veri hacminin artması ile birlikte bu veriden anlamlı bilgilerin üretilmesi de zorlaşmaktadır. Dijital ortamda bulunan veriler kurumlar için birçok alanda bilgi üretilmesini sağlamaktadır. Bu durum hem kar amacı ile kurulan şirketler hem de kar amacı olmayan kamu ya da özel kurumlar için geçerlidir. Şirketler sahip oldukları veriyi kullanarak piyasalarında rekabet avantajı getirebilecek analizler yapar. Birçok şirket verinin bu şekilde kullanılabilmesi amacıyla CRM ve veri madenciliği birimleri oluşturmaktadır. Ayrıca, yasal otoritelerin şirketlerden istediği bilgilerin sunulmasında da yine bu veriler kullanılır. Kar amacı olmayan kurumlar için de veri oldukça önemlidir. İklim modellemesi yapan bir araştırma şirketi için biriktirdiği veriler kritik öneme sahiptir.

Operasyonel sistemlerde bulunan veriler ile üretilen bilginin anlamlı ve tutarlı olması gerekmektedir. Bu nedenle veriden bilgiye giden dönüşümün doğru yapılabilmesi için veri ambarı geliştirimi kendi prensiplerini oluşturmuş ve geleneksel veri ambarı geliştirim yöntemleri ortaya çıkmıştır. Fakat veri ambarı projelerinin hedeflenen başarı düzeyine ulaşamadığı sıklıkla görülmektedir. Bu çalışma ile geleneksel veri ambarı geliştirim yöntemi incelenecek ve iyileştirilebilecek noktalarla ilgili öneriler getirilecektir. Veri ambarı geliştiriminin daha dinamik olarak nasıl yapılabileceği sorgulanmaktadır.

Yöntem olarak veri ambarı geliştirim aşamaları, karşılaşılan sorunlar ve çözüm önerileri ile ilgili kaynak taraması yapılmıştır. Çalışmanın son bölümünde örnek bir uygulama incelenerek bulgular bu uygulama ile desteklenmiştir.

Anahtar Kelimeler: Veri Ambarı, Dinamik Veri Ambarı, İş Zekası

DYNAMIC DATA WAREHOUSE DESIGN AND A SAMPLE APPLICATION

Presented By: Aytaç Bora KUTLU

ABSTRACT

Information is an important asset of institutes and it contributes to the intellectual capital. Data which is created and stored in the digital environment helps to reach information. Regarding the level of technology and its high developing speed, the volume of data has reached to huge amount and will continue to increase for coming years. It is harder to produce meaningful information from data as the data becomes huge. Data enables to produce lots of information for institutes. This is valid for all institutes regardless of being a profit or non-profit institute. Most of the companies use data to gain a market competitive advantage. Companies have CRM and data mining departments to achieve it. Also, companies use the data to provide information to legal authority for their requests. For non-profit institutes, data is crucial also. For a research institute working for climate modeling, the data stored in years is so important.

Information which is derived from data in the operational systems should be meaningful and consistent. In order to transform data to information correctly, data warehousing constituted its own principals and traditional data warehouse development methods emerged. It is frequently seen that data warehouse projects are not successful as expected. With the thesis work, traditional development method is investigated and some proposal are made for potential improvement areas. How to implement a more dynamic data warehouse development is researched.

As a research method, data warehouse development life cycle, potential problems and solutions are examined by literature review. In the final part of thesis, an example of dynamic data warehouse application is described to support the finding and proposals.

Key Words: Data Warehouse, Dynamic Data Warehouse, Business Intelligence

İÇİNDEKİLER

	Sayfa No.
ÖZET	i
ABSTRACT	ii
TABLolar LİSTESİ	v
ŞEKİLLER LİSTESİ	vi
KISALTMALAR	viii
1. GİRİŞ	1
1.1 Problemin Tanımı ve Çalışmanın Amacı	1
1.2 Çalışma Hakkında.....	1
2. VERİ AMBARI	3
2.1 Veri Ambarı Nedir?	3
2.2 Veri Ambarının Tarihsel Gelişimi	4
2.3 Veri Ambarının Amacı	5
2.4 Veri Ambarı Sistem Mimarisi	5
2.5 Veri Ambarı Başarı Ölçütleri.....	7
3. GELENEKSEL VERİ AMBARI GELİŞTİRİMİ	9
3.1 Proje Planı.....	10
3.1.1 Proje Hedef, Kapsam ve Bütçesinin Belirlenmesi	10
3.1.2 Rol ve Sorumluluklarının Belirlenmesi.....	11
3.2 İş İhtiyaçlarının Belirlenmesi	12
3.3 Veri Ambarı Tasarımı, Veri Analizi ve Veri Modelleme.....	13
3.3.1 Veri Ambarı Tasarımı	13
3.3.2 Veri Analizi	15
3.3.3 Veri Modelleme.....	16
3.4 Veri Ambarı Geliştirimi.....	19

3.4.1 Veri Taşıma (Extract).....	20
3.4.2 Veri Dönüştürme (Transformation)	21
3.4.3 Veri Yükleme (Load)	22
3.4.4 Veri Sunum Katmanı Geliştirimi	25
3.4.5 İş Zekası ve Veri Analiz Uygulamaları.....	26
3.5 Veri Ambarı Testleri.....	26
3.6 Üretim Ortamına Geçiş.....	27
4. DİNAMİK VERİ AMBARI TASARIMI.....	29
4.1 Geleneksel Yaklaşımda Karşılaşılan Sorunlar.....	30
4.1.1 Veri Modelleme.....	30
4.1.2 Metadata Yönetimi	34
4.1.3 Veri Sunum Katmanı ve İş Zekası Uygulamaları Entegrasyonu	37
4.2 Geleneksel Yaklaşımda Karşılaşılan Sorunlara Dinamik Yaklaşım Çözümleri	39
4.2.1 Veri Modelleme.....	39
4.2.2 Metadata Yönetimi	41
4.2.3 Veri Sunum Katmanı ve İş Zekası Uygulamaları Entegrasyonu	43
5. ÖRNEK BİR DİNAMİK VERİ AMBARI UYGULAMA İNCELEMESİ.....	45
5.1 Kalido Hakkında Genel Bilgilendirme	45
5.2 Geleneksel Yaklaşımda Karşılaşılan Sorunların Kalido ile Geliştirimi	47
6. SONUÇ.....	58
KAYNAKLAR	60

TABLULAR LİSTESİ

	Sayfa No.
Tablo.1. Veri Ambarı Projelerinin Diğer Yazılım Projelerinden Farkları	9
Tablo.2. Veri Ambarı Projesi Rol ve Sorumlulukları.....	12
Tablo.3. Kimball ve Inmon Yaklaşımları Karşılaştırması.....	15
Tablo.4. Veri Ambarı Kullanım Amacı ve Veri Niteliği.....	38
Tablo.5. Kalido Yaklaşımının Geleneksel Yaklaşımından Farkları.....	47

ŞEKİLLER LİSTESİ

	Sayfa No.
Şekil.1. İş Zekası Piyasa Büyüklüğü	4
Şekil.2. Veri Ambarı Sistem Mimarisi	7
Şekil.3. Veri Ambarı Geliştirim Yaşam Döngüsü	10
Şekil.4. Inmon Yaklaşımı	14
Şekil.5. Kimball Yaklaşımı.....	14
Şekil.6. Kavramsal Model Örneği	17
Şekil.7. Mantıksal Model Örneği.....	18
Şekil.8. Fiziksel Model Örneği	19
Şekil.9. Veri İlişkisi Örneği	23
Şekil.10. Veri Yükleme Yöntemleri	25
Şekil.11. Kavramsal Model Örneği	31
Şekil.12. Fiziksel Model Örneği	32
Şekil.13. İş İhtiyaçlarının Metadata Etkisi	35
Şekil.14. BIM İle Geliştirilmiş Basit Bir Model.....	48
Şekil.15. BIM Modelleme Ekranı-1.....	49
Şekil.16. BIM Modelleme Ekranı-2.....	49
Şekil.17. BIM Fiziksel Veri Ambarı Oluşturma (Deploy) Ekranı.....	50
Şekil.18. BIM Fiziksel Veri Ambarı Oluşturma (Deploy) Sonuç Ekranı.....	50
Şekil.19. MDM Ekran Görüntüsü.....	51
Şekil.20. DIW Ekran Görüntüsü.....	52
Şekil.21. Query Definition Geliştirim Ekranı-1	52
Şekil.22. Query Definition Geliştirim Ekranı-2.....	53
Şekil.23. Query Definition Geliştirim Ekranı-3.....	53
Şekil.24. Query Definition İle Üretilen Rapor.....	54
Şekil.25. Query Definition İle Üretilen SQL Sorgusu.....	54
Şekil.26. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-1	55

Şekil.27. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-2.....	55
Şekil.28. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-3.....	56
Şekil.29. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-4.....	56
Şekil.30. İş Zekası Uygulamasında Oluşturulan Veri Modeli	57



KISALTMALAR

CRM	: Customer Relationship Management
ETL	: Extract – Transform - Load
3NF	: Third Normal Form
DDW	: Dynamic Data Warehouse
BIM	: Business Information Modeler
DIW	: Dynamic Information Warehouse
MDM	: Master Data Management
UID	: Universal Information Director

1. GİRİŞ

Kurumlar sahip oldukları veriyi farklı şekillerde kullanarak kendilerine değer katmayı hedeflerler. Ulaşılmak istenen bilgiye göre veri işlenmeden ham hali ile kullanılır ya da verinin işlenmesiyle istenen bilgi üretilir. Örneğin, günlük olarak üretim miktarını raporlamak için veri işlemeye gerek duyulmaz. Operasyonel sistemde bulunan veri doğrudan kullanılarak bu bilgi üretilir. Fakat günlük üretim miktarlarını karşılaştıran, kaynak verimliliğinin hesaplandığı bir raporlama için verinin işlenmesi gerekir. Her iki senaryoda da operasyonel sistemlerde bulunan verinin bir şekilde kullanılması gerekmektedir. Operasyonel sistemlerde bulunan veriye doğrudan erişerek bu bilgilerin üretilmeye çalışılması hem operasyonel sistemler üzerinde hem üretilen bilginin oluşturulmasında sorunlara yol açmaktadır. Bu sorunların önüne geçebilmek için veri ambarları oluşturulmaktadır.

1.1 Problemin Tanımı ve Çalışmanın Amacı

Teknolojik gelişmeler neticesinde dijital ortamda oluşan ham veri miktarı her geçen gün artmaktadır. Ham veriler işlenerek bilgiye dönüştürülür. Üretilen bilgiler ise karar verme, mevcut durumu izleme ve geleceğe yönelik tahminler yürütülmesinde kullanılır.

Kurumlar için sahip oldukları veri dijital sermayelerinin önemli bir bölümünü oluşturmaktadır. Veri hacminin artmasıyla birlikte verinin bilgiye dönüştürülmesinin önemi ve zorluğu da artmaktadır. Bu nedenle, sahip olunan verinin anlamlı ve tutarlı bilgilere dönüştürülmesi kurumlar için önemli bir hedef haline gelmiştir. Belirtilen hedefe ulaşabilmek için veri ambarı projeleri hayata geçirilmektedir. Fakat veri ambarı projelerinin büyük çoğunluğu başarısız ya da sınırlı seviyede başarılı olarak değerlendirilmektedir [1]. Çalışma ile veri ambarı projelerinin başarısız ya da sınırlı seviyede başarılı olarak nitelendirilmesine sebep olan faktörlerin incelenmesi amaçlanmıştır.

1.1 Çalışma Hakkında

Tez çalışması tanımlanan problem ve amaca göre organize edilmiş 4 temel bölümden oluşmaktadır.

İlk olarak 2 numaralı bölümde veri ambarı ile ilgili genel bilgilendirme sunulmuştur. Veri ambarı nedir, tarihsel gelişimi nasıl olmuştur, kullanım alanları nelerdir gibi konularda bilgi verilmiştir. Bu bölümde son olarak veri ambarı projelerinin başarı ölçütlerine değinilmiştir.

Çalışmanın 3 numaralı bölümünde geleneksel veri ambarı geliştirim yöntemi hakkında literatür tarama çalışması yapılarak geliştiriminde uygulanan geleneksel yöntemler detaylı olarak anlatılmıştır.

Geleneksel veri ambarı geliştirim yönteminin aktarıldığı bölümden sonra, çalışmanın 4 numaralı bölümünde, dinamik veri ambarı tasarımı konusu işlenmiştir. Bu bölümde, dinamik veri ambarı tasarım yaklaşımı hakkında bilgilendirme, geleneksel veri ambarı geliştirim yönteminde iyileştirilebilecek kısımlar ve dinamik veri ambarı tasarım yaklaşımı ile çözüm önerileri aktarılmıştır.

Dinamik veri ambarı yaklaşımı ile getirilen çözüm önerilerini destekleyecek örnek bir uygulama 5 numaralı bölümde incelenmiştir.

Son olarak; 6 numaralı sonuç bölümünde çalışmanın özeti yapılarak dinamik veri ambarı tasarımının temel prensiplerine değinilmiştir.

2. VERİ AMBARI

2.1 Veri Ambarı Nedir?

Veri sorgulama ve veri analizi için tasarlanmış ilişkisel veri tabanına veri ambarı denir. Veri ambarında birçok farklı kaynaktan gelen veri bulunmaktadır. Bu veriler sadece güncel veriyi değil, ayrıca tarihsel veriyi de barındırır. Veri ambarları birçok farklı kaynaktan gelen verinin birleştirilmesini ve veri analiz işinin kaynak sistemlerden ayrılmasını sağlar. Veri ambarı ortamlarında ilişkisel bir veri tabanına ek olarak, verinin birleştirilmesi için veri taşıma (E-Extract), veri dönüştürme (T-Transform), veri yükleme (L-Load) çözümü (ETL), verinin son kullanıcıya sunulması için veri görüntüleme ve işleme uygulamaları bulunur [2].

Veri ambarı tek başına bir uygulama ya da ürün değildir. Veri ambarı oluşturulması ve sonrasında sürdürülebilmesi için veri ambarı ortamında birçok farklı teknoloji ve uygulama kullanılması gerekmektedir. Veri ambarında bulunan verilerin karakteristik özellikleri aşağıdaki gibi tanımlanmaktadır [3].

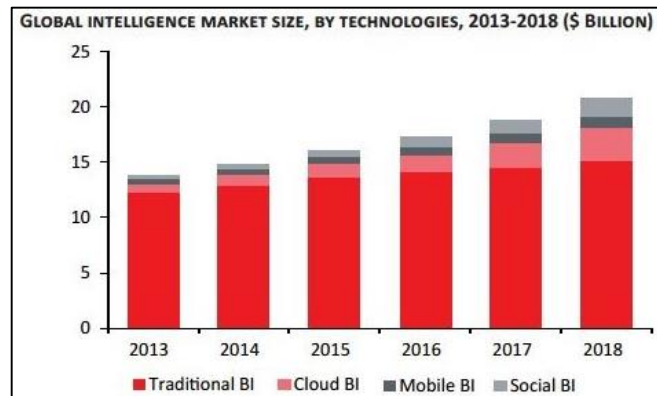
- **İş Alanına Yönelik (Subject Oriented):** Veri ambarları verinin analizini kolaylaştırmak için tasarlanırlar. Verinin ait olduğu iş alanına ve iş ihtiyaçlarına göre veri ambarı tasarımları yapılmalıdır. Ticari ürünler satan bir kurumda satış verisinin tutulduğu birçok operasyonel sistem olabilir. Veri ambarında ise bu sistemlerdeki veri satış, müşteri, ürün gibi iş alanına yönelik bilgileri tutacak şekilde oluşturulmalıdır.
- **Bütünlük (Integrated):** Veri ambarına birçok farklı kaynaktan veri gelmektedir. Farklı kaynaklardan gelen verinin bir bütünün parçası olacak şekilde birbiriyle uyumlu olarak birleştirilmesini sağlamak oldukça önemlidir. Aynı bilgiyi ifade eden veriler farklı sistemlerde farklı değerler ile saklanabilir. Örneğin bir ürün için 2 farklı sistemde farklı ürün kodları kullanılmış olabilir. Ürünlerle ilgili veriyi veri ambarına taşıırken ürün kodu standart bir değere dönüştürülmelidir. Bu şekilde farklı sistemlerde bulunan verinin birbiri ile uyumlu olarak birleştirilmesi sağlanmış olur.
- **Kalıcılık (NonVolatile):** Verinin kalıcılık kavramı veri ambarına giren değerlerin daha sonra değişmediğinin ifadesidir. Verinin tüm yaşam döngüsünü takip edebilmek için veri ambarına giren bir verinin güncellenmemesi gerekir. Eğer

bir veriye ait daha güncel bir değer gelirse veri ambarında var olan veriyi güncellemek yerine yeni veri ayrıca kayıtlanmalıdır. Her iki bilginin ait olduğu zaman bilgisi de tutulduğu için böylelikle tarihsel bir verinin oluşması sağlanır.

- **Zamana Bağımlılık (Time Variance):** Veri ambarına giren tüm verilerin ait olduğu zaman dilimini gösteren bir zaman bilgisinin olması gerekmektedir. Böylelikle verinin zaman içerisindeki değişimi incelenerek eğilim analizleri yapılır.

2.2 Veri Ambarının Tarihsel Gelişimi

1970’li yıllarda bilgisayar donanımı pahalı ve bilgisayarların işlemci güçleri zayıftı. Bu yüzden dijital ortamda bulunan veri sınırlıydı ve verinin farklı amaçlar için kullanılması henüz yaygınlaşmamıştı. 1980’li yıllarda ilişkisel veri tabanları ortaya çıktı. Veriler birbiri ile ilişkili tablolar üzerinde tutulmaya başlandı. 1990 ve sonrasında teknolojinin hızla gelişmesi ile birlikte artık bilgisayarlar bireysel kullanıcıların evlerine kadar girmeye başladı. Bu durum dijital ortamda bulunan verinin hızla artmasına yol açtı ve iş zekası uygulamalarına olan ihtiyaçlar artmaya başladı [4]. İş zekası ihtiyaçlarını karşılayabilmek için veriyi bir merkezde birleştirerek veri ambarları oluşturulmaya başlandı. 2000 ve sonrasında ise dijital ortamda bulunan verinin önemi ve hacmi artmaya devam etti. Veri ambarı ve iş zekası uygulamalarının önemi daha fazla anlaşıldı. Bunun sonucunda, birçok kurum sahip olduğu veriyi kullanabileceği veri ambarı ve iş zekası sistemleri geliştirdi. Teknolojik ilerlemelere paralel olarak artan veri hacmi ve verinin kullanım alanları sonucu veri ile ilgili sistemlere yapılan yatırımlar da artacaktır. Şekil.1. ile iş zekası piyasa büyüklüğünün önümüzdeki yıllarda artmaya devam edeceği görülmektedir. [5].



Şekil.1. İş Zekası Piyasa Büyüklüğü

2.3 Veri Ambarının Amacı

Kurumlar sahip oldukları veriyi kullanarak mevcut durum analizi, eğilim analizi, yasal raporlamalar, finansal analiz, geleceğe dönük tahminler ve benzeri analizler yapmayı hedefler. Verinin farklı amaçlar için kullanılabilmesi için veri ambarına ihtiyaç duyulur. Veri ambarının amacı veri üzerinden yapılacak analizlerde her zaman kabul gören bir kaynak olmaktır. Bunun için veri ambarlarında aşağıdaki niteliklerin olması amaçlanır [6].

- Veri ambarı kolay erişilebilir olmalıdır. Veri ambarının içerdiği bilgi anlaşılabilir olmalıdır. Alan isimlendirmeleri bunu sağlayacak şekilde oluşturulmalı, veri ambarına erişim sağlayan uygulamalar ile veriye hızlı ve kolay bir şekilde erişilmelidir.
- Veri ambarı kendi içinde tutarlı olmalıdır. Veri ambarında bulunan her bir bilginin tanımı ve kaynağı bellidir. Veri ambarında birbiri ile ilişkili bilgiler çelişmemelidir. Örneğin, müşteri tablosunda bulunan son satış tarihi bilgisi ile satış tablosundaki veriler birbiri ile uyumlu olmalıdır.
- Veri ambarı değişimlere çabuk uyum sağlamalıdır. İş ihtiyaçları gereği veri ambarına yeni bilgiler eklenmesi gerektiğinde mevcut veri etkilenmeden yeni alan eklenmeli, veri ambarının iş ihtiyaçlarına her zaman cevap vermesi sağlanmalıdır.
- Veri ambarı değişiklik taleplerini hızlı bir şekilde gerçekleştirmelidir.
- Veri ambarı bilgi güvenliğini gözetmelidir. Veri ambarı kullanıcıların sadece kendi iş alanları ile ilgili veriye erişmesini sağlamalıdır.
- Veri ambarı karar verme analizleri için gerekli tüm veriyi içermelidir.
- Veri ambarı son kullanıcılar tarafından başarılı bir uygulama olarak değerlendirilmeli ve aktif olarak kullanılmalıdır.

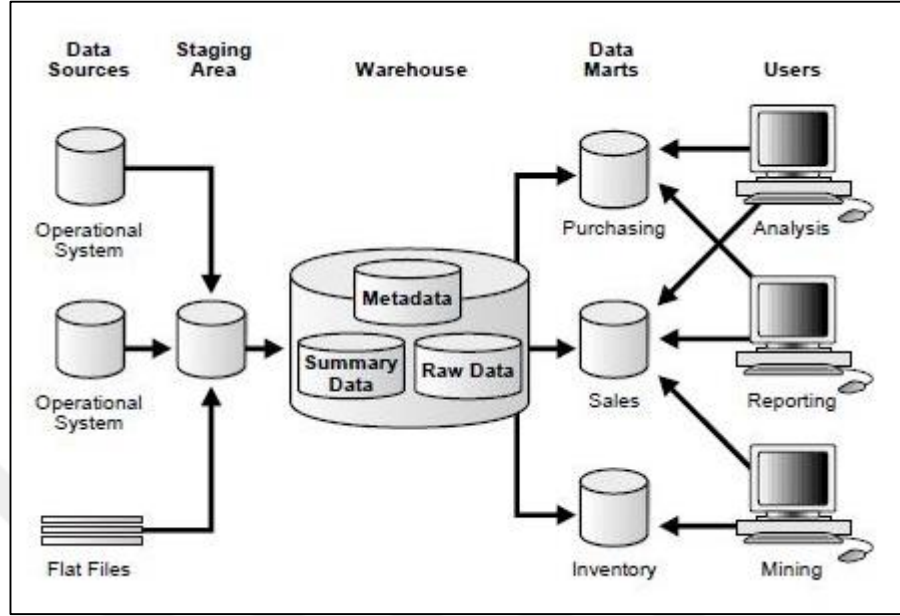
2.4 Veri Ambarı Sistem Mimarisi

Veri ambarı bir ya da daha fazla kaynaktan alınan ilişkili verileri birleştiren ve analiz edilmesini sağlayarak iş zekasını arttıran bir teknolojidir [7]. Veri ambarı tek bir uygulama ya da ürün değildir, içerisinde birçok uygulama ve ürünü barındıran, birbiriyle bütünleşmiş bir sistem mimarisidir. Veri ambarları ve mimarileri kurumların

iş ihtiyaçlarına bağlı olarak değişebilmektedir. Veri ambarı mimarisinde bulunan bileşenler Şekil.2. ile gösterilmektedir.

- **Veri Kaynakları:** Veri ambarına alınarak işlenecek verilerin ait olduğu yapıları ifade eder. En büyük veri kaynakları kurumlara ait operasyonel sistemlerdir. Ayrıca, kuruma ait olmayan sistemler de veri kaynağı olabilir. Örneğin, bir banka için merkez bankası döviz kurları kuruma ait olmayan bir veri kaynağıdır. Veri kaynakları veri tabanlarından farklı ortamlarda da veri tutabilir. Örneğin, .txt ya da csv formatında dosyalar da bir veri kaynağı olabilir.
- **Veri Ara Katmanı: (Staging Area)** Verinin depolandığı ilk katmandır. Bu katmanda veri temizleme (data cleaning), veri dönüştürme (data transformation), veriyi standart hale getirme (data standardization) gibi işlemler yapılır. Ayrıca, farklı sistemlerden gelen veri aynı veri ortamında bulunduğu için veriyi birleştirme işlemleri daha hızlı yapılabilir. Veri ara katmanı kaynak sistemlerdeki verinin çoklanmasına neden olmakta ve veri depolama maliyetlerini arttırmaktadır. Bu nedenle iş ihtiyaçlarına göre veri ara katmanı olmayan mimari tasarımlar da yapılmaktadır.
- **İşlenmiş Veri Katmanı (Warehouse):** Kaynak sistemden gelen verilerin işlenerek depolandığı katmandır. Metadata, ham veri ve özet veri olarak veriler depolanır. Son kullanıcılar ve veri analiz uygulamaları veri ambarı tasarımına göre doğrudan bu katmana erişerek ya da işlenmiş veri kullanılarak üretilen data martlar üzerinden veriye erişir.
- **Veri Gösterim Katmanı: (Presentation Layer)** İşlenmiş veri kullanılarak üretilen data mart ve diğer veri tabanı nesnelere gösterildiği katmandır. Son kullanıcılar ve veri analiz uygulamaları veriye bu katman üzerinden erişir. İşlenmiş veri katmanında normalize bir veri modeli ile saklanan veri denormalize edilerek iş alanına yönelik (subject oriented) data martlar oluşturulur. Data martlar ile veriye daha hızlı erişilmesi ve kullanıcıların sadece kendileri ile ilgili veriye erişmesi sağlanır.

- **Veri Erişim ve İşleme Uygulamaları:** İşlenmiş veriye erişimi sağlayarak raporlama, analiz ve veri madenciliği çalışmalarının yapılması sağlanır.



Şekil.2. Veri Ambarı Sistem Mimarisi

Kaynak: Lane, P., *Oracle9i Data Warehousing Guide Release 2 (9.2)*,
https://docs.oracle.com/cd/B10500_01/server.920/a96520/concept.htm#51078, 2002.

2.5 Veri Ambarı Başarı Ölçütleri

Veri ambarının başarı değerlendirmesi tek seferlik bir değerlendirme değildir ve uygulamada kaldığı sürece devam eder. Bu nedenle veri ambarının başarısı zaman içerisinde değişebilir. Veri ambarı, en basit tanımıyla, ne kadar sık kullanılıyorsa o kadar başarılıdır [8]. Kullanıcıların iş ihtiyaçlarına cevap verdiği sürece veri ambarı kullanılmaya ve gelişmeye devam eder. Kullanıcı iş ihtiyaçlarını karşılayamayan veri ambarının kullanımı sıklığı düşer ve veri ambarı başarısız olarak değerlendirilmeye başlar. Veri ambarının içerdiği veri temiz, tutarlı, yeterli, erişilebilir ve güncel olduğu sürece veri ambarı kullanılır. Verinin bu niteliklere sahip olması için veri ambarı sisteminin de belirli şartları sağlaması gerekir. Veri ambarı sistem performansı kullanıcı beklentilerini karşılamalıdır. Ayrıca, kullanıcı ihtiyaçlarının hızlıca uyarlanabildiği bir sistem olmalıdır. Kullanıcı ihtiyaçları zaman içerisinde değişir ve

gelişir. Veri ambarı kullanıcı ihtiyaçlarına paralel olarak gelişemezse güncelliğini kaybeder. Bu durum veri ambarının kullanımını ve dolayısıyla başarı değerlendirmesini etkiler.



3. GELENEKSEL VERİ AMBARI GELİŞTİRİMİ

Ana fonksiyonlarına göre bir veri ambarı 3 bileşenden oluşur. Bu bileşenler veri oluşturma, veri saklama ve veri sunum fonksiyonlarıdır. Veri ambarı projeleri diğer operasyonel sistem projelerinden bu 3 ana fonksiyon alanında farklılaşır. Tablo.1. ile veri ambarı projelerinde bulunan farklılıklar gösterilmektedir [9].

Tablo.1. Veri Ambarı Projelerinin Diğer Yazılım Projelerinden Farkları

Veri Oluşturma	Veri Saklama	Veri Sunum
Çok sayıda ve farklı sistemden kaynak	Çok büyük hacimli verilerin saklanması	Farklı kullanıcı tipleri
Farklı teknolojik platformlar	Verinin hızlı büyümesi	Farklı sorgu tipleri
İlk yüklemde taşınan veri büyüklüğü	Yeni veri tipleri saklama	Çok boyutlu analiz
Düzenli veri taşıma işlemi	Veri arşivleme	OLAP kullanımı
Veri saklama maliyeti	Veri erişim performans kaygısı	Metadata yönetimi
Veri entegrasyon zorluğu	İş zekası uygulamaları ile uyumluluk	Veri madenciliği kullanımı
Karışık veri dönüştürme ve temizleme işlemleri	Ara katmanlarda veri depolama	Farklı iş zekası uygulamalarının kullanılması

Yazılım projelerinde uygulanan geliştirim yaşam döngüsünün veri ambarı projelerinin ihtiyaçlarına göre uyarlanması gerekir [9]. Veri ambarı projelerinde uygulanan geliştirim yaşam döngüsü Şekil.3. ile gösterilmiştir [10].



Şekil.3. Veri Ambarı Geliştirim Yaşam Döngüsü

3.1 Proje Planı

Bir projenin başarılı bir şekilde hayata geçirilebilmesi için etkin bir planlama ve proje yönetimi yapılmalıdır. Projenin başarısızlıkla sonuçlanmasına en fazla etki eden etken uygun olmayan planlama ve yetersiz proje yönetimidir [9]. Bu nedenle projenin hedef, kapsam, bütçe ve planlaması net olarak ortaya konmalı ve projenin tüm aşamalarında takip edilmelidir.

3.1.1 Proje Hedef, Kapsam ve Bütçesinin Belirlenmesi

Veri ambarı projesine neden ihtiyaç duyulduğu ilgili iş kolları ve proje destekleyicileri ile birlikte değerlendirilmelidir. Veri ambarı tasarımının en doğru çözüm olduğu konusunda mutabakat sağlanmalıdır. Böylelikle üst yönetimin de projeyi desteklemesi sağlanır. Üst yönetim desteği olmadan yapılan büyük projelerin başarılı olması mümkün değildir [9].

Proje ile ulaşılmak istenen hedefler açık olarak belirtilmelidir. Veri ambarının kullanılması hedeflenen fonksiyonlar, sağlayacağı kar/fayda, oluşacak maliyet azalması gibi hedefler açıkça tarif edilmelidir. Böylelikle projede çalışan ve proje

destekçilerinin ortak bir hedefe odaklanmaları ve proje tamamlandığında yapılan proje başarı değerlendirilmesinde hedeflenen ve gerçekleşen durum karşılaştırılması sağlanır.

Projenin hedeflerine ulaşabilmesi için proje kapsamının net olarak belirlenmesi gerekir. Proje kapsam ve hedeflerine göre proje planlaması yapılır. Projeye başlandıktan sonra kapsamda değişiklik yapılması daha önce yapılmış tüm planlamaları etkileyecek ve projenin başarılı bir şekilde tamamlanmasını riske atacaktır. Proje kapsamına göre aşağıdaki unsurlara karar verilir ve projenin bu planlamalar çerçevesinde yürütülmesi planlanır.

- Projede ihtiyaç duyulan insan ve teknoloji kaynağının belirlenmesi
- Projede yer alacak işlerin belirlenmesi
- Proje bütçesinin belirlenmesi

Kapsam ve bütçe birbirini karşılıklı etkileyen unsurlardır. Kurumlar kapsamı belirledikten sonra bütçe çalışması yapabildikleri gibi, bir bütçe belirledikten sonra da kapsam çalışması yapılabilir.

3.1.2 Rol ve Sorumluluklarının Belirlenmesi

Veri ambarı tek başına bir uygulama ya da ürün değildir. Veri ambarı ortamında birden fazla uygulama ve teknoloji bulunur. Ayrıca, veri ambarı projelerinin diğer yazılım projelerinden farklı olduğu unsurlar bulunmaktadır [9]. Bu sebeplerden dolayı veri ambarı projelerinde çok çeşitli uzmanlık alanlarına sahip insan kaynağına ihtiyaç duyulur.

Proje planındaki her bir işten hangi rollerin sorumluluğu olduğu net olarak belirlenmeli ve işlerin tamamlanma durumu sorumluluk listesine göre belirlenmelidir. Veri ambarı projelerinde yer alabilecek roller ve sorumlulukları Tablo.2. ile gösterilmiştir. Projenin niteliklerine göre bazı rollere gerek olmayabilir.

Tablo.2. Veri Ambarı Projesi Rol ve Sorumlulukları

Roller ve Sorumluluklar	
Üst Yönetim Destekleyicisi Yönetim, destek, karar verme	Veri Tabanı Yöneticisi Veri tabanı operasyonları
Proje Yöneticisi Kontrol, takip, iş atama	Veri Dönüşüm Uzmanı Veri taşıma, birleştirme, dönüştürme
İş kolu Yöneticisi Kullanıcı koordinasyon	Kalite Kontrol Analisti Veri kalite kontrolü
Sistem Mimarı Mimari tasarım	Test Koordinatörü Veri geliştirim testleri ve yönetimi
Teknik Altyapı Uzmanı Teknik altyapı kurulumu	Yazılım Uzmanı Kod geliştirme
İş Analisti İhtiyaç Analizi	Son Kullanıcı Uygulamaları Uzmanı Uygulama entegrasyonu
Veri Modelleme Uzmanı Modelleme	Eğitim Sorumlusu Eğitim faaliyetleri

Kaynak: Ponniah, P. (2001), *Data Warehouse Fundamentals / A Comprehensive Guide for IT Professionals*, s. 78, Wiley Publications, USA.

3.2 İş İhtiyaçlarının Belirlenmesi

Projenin belirlenen hedefleri gerçekleştirebilmesi için veri ambarının çözüm sağlayacağı iş fonksiyonlarının belirlenmesi gerekir. Proje kapsamına göre belirlenen iş fonksiyonları ve çözüm yöntemleri üzerinde proje ekibi ve proje destekleyicileri hem fikir olmalıdır. Bu şekilde tüm proje ekibi tarafından iş ihtiyaçlarının doğru olarak anlaşılması ve doğru çözüm yöntemlerinin geliştirilmesi sağlanır.

Veri ambarı projesinin başarılı olabilmesi için iş fonksiyonlarını bilen kullanıcılarının proje ekibinde yer alması önemlidir. İş fonksiyonları konusunda yetersiz ve yanlış bilgilendirme olması durumunda proje tamamlandığında ortaya çıkacak ürün iş ihtiyaçlarını karşılamayacaktır. Veri ambarı geliştiriminin başarılı bir ürün olma ihtimali iş ihtiyaçlarını ve son kullanıcıları daha iyi anlayarak artırılır [11].

İş ihtiyaçlarının belirlenmesi ile veri ambarından fonksiyonel ve teknik kapasite anlamında neler beklendiği genel hatlarıyla ortaya çıkar. Bu aşamada aşağıdaki konularda bilgi sahibi olunur ve projenin ileri aşamalarında detay analizler buna göre gerçekleştirilir.

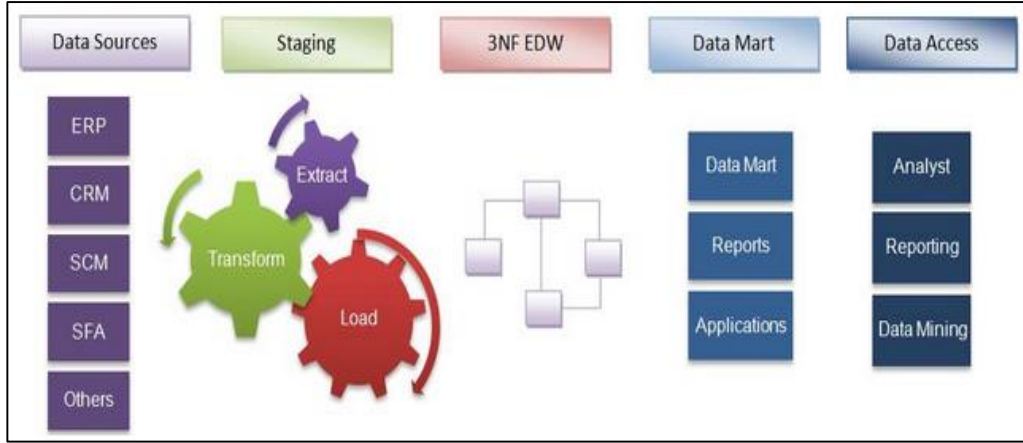
- Ölçümlenmesi gereken sayısal bilgiler (Satış adedi, üretim miktarı gibi)
- Ölçümlenmesi gereken bilgilerin sahip olması gereken boyutlar (Müşteri, şube, zaman, ürün gibi)
- Veri ambarına alınacak verilerin ait olduğu sistemler
- Veri ambarının hangi sıklıkla güncellenmesi gerektiği (Günlük, haftalık, aylık)
- Veri ambarı kullanıcı tipleri ve adedi
- Veriye ne sıklıkla ve ne şekilde erişilmesi gerektiği
- Geriye dönük veri ihtiyacı (1 yıl, 5 yıl, 10 yıl gibi)
- Veri ambarı sorgu cevap beklenti süresi

3.3 Veri Ambarı Tasarımı, Veri Analizi ve Veri Modelleme

3.3.1 Veri Ambarı Tasarımı

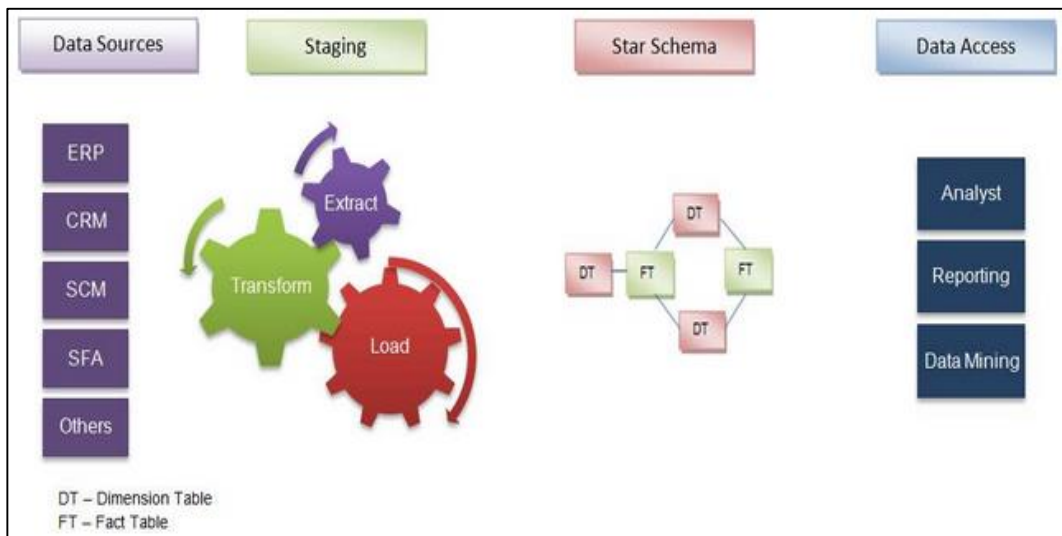
Veri ambarı tasarımında yaygın olarak kullanılan 2 yaklaşım bulunmaktadır. Bu yaklaşımlar Bill Inmon ve Ralph Kimball tarafından ortaya konmuştur. Hangi yaklaşımın daha iyi olduğu sorusunun kesin bir cevabı yoktur çünkü her iki yaklaşımın da kendine göre avantajları ve farklılaştığı unsurlar bulunmaktadır [12]. İki yaklaşımın güçlü olduğu yönleri kullanarak hibrit bir tasarım yapmak da mümkündür.

- **Bill Inmon Yaklaşımı:** Bill Inmon'a göre veri ambarı tüm iş zekası sisteminin bir parçasıdır ve her kurumun bir adet veri ambarı bulunur [13]. Inmon'ın yaklaşımında yukarıdan aşağıya (top-down approach) giden bir tasarım uygulanır. Veri ambarı kaynak sistemlerden aldığı veriyi normalize bir model ile (3NF) saklayan merkezi bir depodur [14]. Tüm data martların oluşturulmasında en üstte bulunan veri ambarı kullanılır. Özet olarak, Inmon yaklaşımını ifade eden yukarıdan aşağıya tasarım ile kaynak sistemlerde bulunan veri normalize olarak modellenmiş bir veri tabanına alınır ve sonrasında ihtiyaç duyulan data martlar bu kaynaktan oluşturulur. Şekil.4. ile Inmon yaklaşımı gösterilmektedir [12].



Şekil.4. Inmon Yaklaşımı

- Ralph Kimball Yaklaşımı:** Kimball'a göre veri ambarı operasyonel sistemlerde bulunan verinin raporlama ve analiz için tasarlanmış kopyasıdır [12]. Kimball'ın yaklaşımında aşağıdan yukarıya (bottom-up approach) giden bir tasarım uygulanır. Veri ambarı denormalize edilmiş yıldız şema (star schema) data martlardan oluşmaktadır [14]. Özet olarak, Kimball yaklaşımını ifade eden aşağıdan yukarıya tasarım ile normalize bir data model ile oluşturulmuş veri ambarı bulunmaz. Veri ambarı doğrudan data martlardan oluşturulur. Şekil.5. ile Kimball yaklaşımı gösterilmektedir [12].



Şekil.5. Kimball Yaklaşımı

Inmon ve Kimball yaklaşımlarının birbirine göre farklı avantajları ve dezavantajları bulunmaktadır. Veri ambarı projesinin ihtiyaçları, hedefleri ve bütçesine göre uygun olan yaklaşım seçilir. Inmon ve Kimball yaklaşımlarının karşılaştırılması Tablo.3. ile sunulmuştur [15].

Tablo.3. Kimball ve Inmon Yaklaşımları Karşılaştırması

Özellik	Kimball Yaklaşımı	Inmon Yaklaşımı
Veri Kullanım İhtiyacı	Taktiksel	Stratejik
Veri Birleştirme İhtiyacı	Spesifik iş alanı bazında	Kurumsal olarak tüm iş alanları
Veri içeriği	İş ihtiyaçlarına göre hesaplanmış bilgiler, performans ölçüm bilgileri	Farklı ve değişebilecek iş ihtiyaçlarını karşılayabilecek alt seviye bilgiler
Veri Ambarı Büyüme Potansiyeli	Büyümü ve farklı ihtiyaçları karşılama potansiyeli düşük	Büyümü ve farklı ihtiyaçları karşılama potansiyeli yüksek
Veri beklentisi	Kaynak sistemlerden gelen veri istikrarlıdır	Kaynak sistemlerden gelen veri değişkendir
İnsan Kaynağı ve Uzmanlık İhtiyacı	Küçük ekipler ve genel uzmanlıklar gerekir	Büyük ekipler ve özel uzmanlıklar gerektirir
Geliştirim Süresi	Kısa	Uzun
Geliştirim Maliyeti	Başlangıç için daha düşük maliyet, sonraki geçişlerde ilk geçiş ile aynı maliyet	Başlangıç için yüksek maliyet, sonraki geçişlerde daha düşük maliyet

3.3.2 Veri Analizi

Veri analizi ile iş ihtiyaçlarına yönelik çözümlerin geliştirilebilmesi için gerekli olan veri ile ilgili tüm bilgilerin toplanması sağlanır. İş ihtiyaçlarının veri ihtiyaçlarına dönüştürüldüğü aşamadır. Geliştirilecek veri ambarı ile ilgili bir temel oluşturulur. Veri modelleme ve geliştirimi bu aşamada ortaya çıkarılan bilgilere göre yapılır. Bu nedenle doğru ve yeterli detayda bilgi içermesi gerekir. Bu aşamada yapılan bir hata projenin birçok geliştirim noktasını etkiler ve projede beklenmeyen sonuçlar oluşturabilir. Öncelikle ihtiyaç duyulan tüm veriler ve bu verilerin ait olduğu varlıklar (tablolar) belirlenir. Böylelikle veri tabanında oluşturulacak tablo ve kolon listesi elde edilmiş olur. Veri analizi ile aşağıdaki bilgiler elde edilir.

- Her bir verinin tanımı (Veri sözlüğü oluşturulmasında da kullanılır.)
- Her bir verinin veri ambarında ait olacağı tablo bilgisi
- Veri özellikleri (Veri tipi, veri uzunluğu, primary key, foreign key gibi bilgiler)
- Standart bir isimlendirme yöntemi kullanılarak oluşturulmuş tablo ve veri isimleri (Örneğin; müşteri numarası bilgisi standart bir isim kullanılarak veri ambarındaki ilgili tablolara eklenmelidir.)
- Verilerin üretilmesi için gerekli olan kaynak sistem tabloları ve kolonları
- Verilerin nasıl hesaplanacağı bilgisi
- Veriler arası hiyerarşik ilişkiler
- Fact ve Dimension tablolar

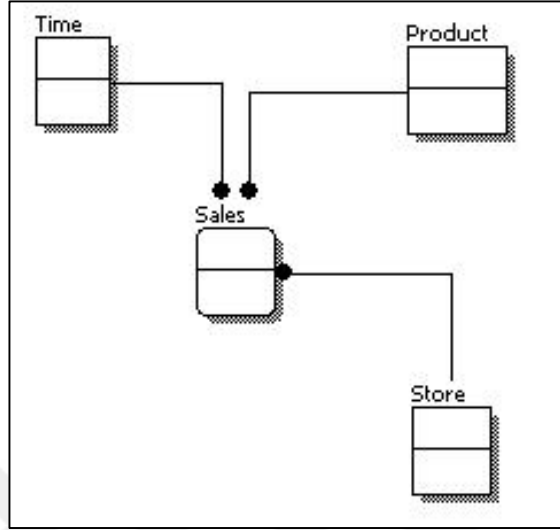
3.3.3 Veri Modelleme

İş süreçlerinin gerçekleştirilebilmesi için operasyonel sistemler geliştirilir. Operasyonel sistemlerde üretilen veri iş süreçlerinin sonucunda oluştuğu için yapılan işle ilgili bilgiye ulaşılmasını sağlar. Veri modeli iş süreçleri sonucunda oluşan verinin düzenlenmiş bir gösterimidir. Bu nedenle, iş süreçlerini anlamamanın ve yönetmenin en iyi yolu veri modellemedir. Veri modeli verinin yapısal ve içerik olarak düzenlenerek veri ambarında depolanmasını sağlar [16]. Veri modelleme veri ambarı tasarımının gerçekleştirilmesine katkıda bulunur. Veri analiz aşamasında oluşturulan bilgiler ile veri modellemesi yapılır. Veri modelleme aşamasına başlamak için veri analiz aşamasının tamamlanmış olması gerekmez. Veri analiz aşamasında veri listesinin belirli bir seviyede oluşması sonrasında veri modellemeye başlanabilir. Bu nedenle veri analizi ve veri modelleme aşamaları birbirine paralel olarak yürütülebilir. Veri modelleme genel olarak üç aşamada gerçekleştirilir.

- **Kavramsal (İlişkisel) Modelleme:** Veri hakkında özet bir bilgi verir. Operasyonel sistemleri kullanan ve tasarlayan kullanıcılar ile birlikte oluşturulur. İş süreçleri ve veri hakkında kavramsal bir fikir oluşturulmasını sağlar. Kavramsal modelin karakteristik özetleri aşağıdaki gibi özetlenebilir [17].

- Veri ambarında oluşturulacak önemli varlıklar ve aralarındaki ilişkiler gösterilir.

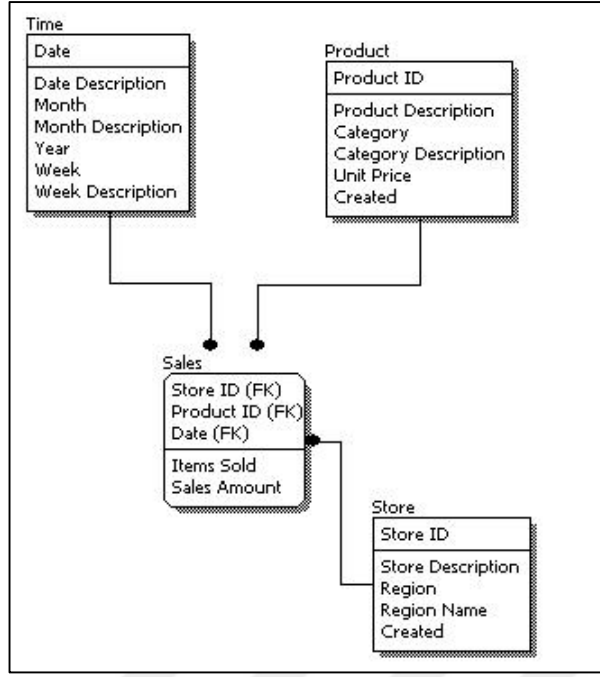
- Varlıklarla ilgili detay bilgi bulunmaz.
- Veri tabanında kullanılacak primary key gibi teknik bilgiler bulunmaz.
- Modellenen varlıklar tanımlanır ve aralarındaki ilişki gösterilir.



Şekil.6. Kavramsal Model Örneği

- **Mantıksal Modelleme:** Veri hakkında olabildiğince fazla bilgi içerir. Veri ambarının geliştiriminin yapılacağı teknoloji ve üründen bağımsız olarak modellenir. Veri ambarının fiziksel olarak oluşturulmadan önce gerekli tüm bilgilerin hazırlandığı aşamadır. Mantıksal modelin karakteristik özellikleri aşağıdaki gibi özetlenebilir [18].

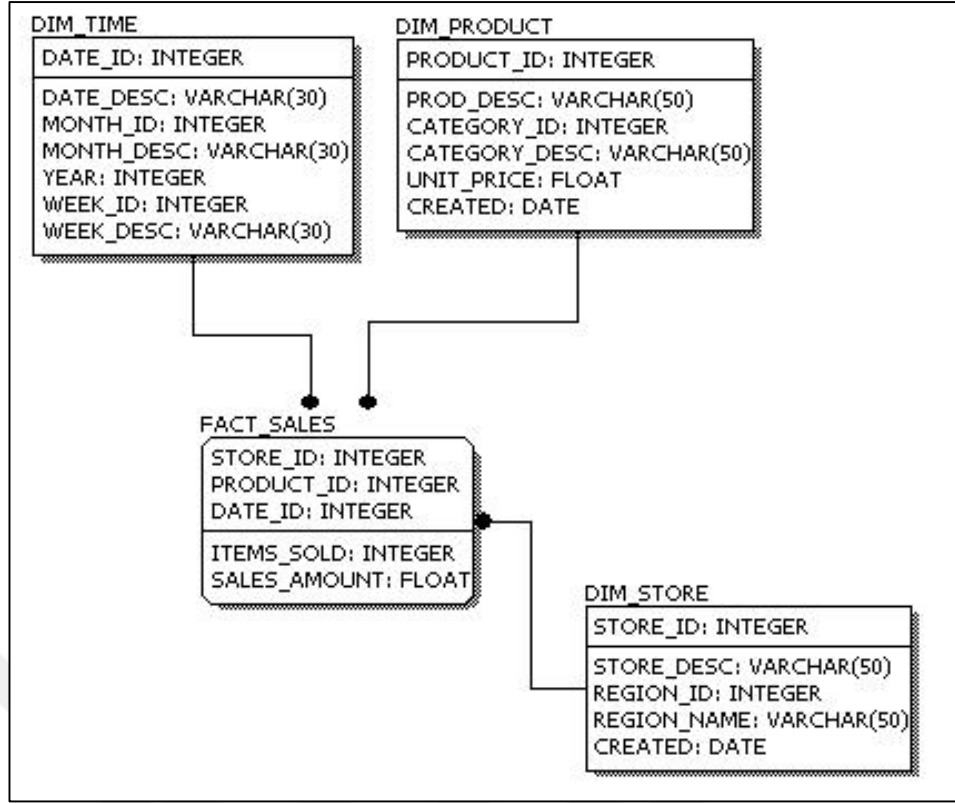
- Veri ambarında oluşturulacak tüm varlıklar ve ilişkileri gösterilir.
- Veri ambarında varlıklar üzerinde tutulacak tüm bilgiler gösterilir.
- Primary ve foreign key bilgileri bulunur.
- Veri ambarı tasarımına göre normalizasyon gerekli ise bu aşamada yapılır.



Şekil.7. Mantıksal Model Örneği

- **Fiziksel Modelleme:** Verinin fiziksel veri tabanı ortamında nasıl oluşturulduğunu gösterir. Veri tabanında oluşturulacak tabloların yapısı detaylı olarak gösterilir. Tablo adı, kolon adı, kolon veri tipi, primary key ve foreign key gibi tüm bilgiler gösterilir. Veri tabanının oluşturulacağı teknoloji ve ürüne göre farklılık gösterir. Fiziksel modelin özellikleri aşağıdaki gibi özetlenebilir [28].

- Veri ambarında bulunacak tüm tablo ve kolon isimleri gösterilir.
- Tablolar arasındaki ilişkiyi göstermek için foreign key kullanılır.
- Veri ambarı tasarımına göre gerekli ise denormalizasyon bu aşamada gerçekleştirilir.
- Fiziksel ortamın oluşturulacağı teknoloji ve ürüne bağlıdır. Örneğin, veri tipinin gösterimi geliştirim yapılacak ürüne göre gösterilmelidir.



Şekil.8. Fiziksel Model Örneği

3.4 Veri Ambarı Geliştirimi

Veri ambarı geliştirimine başlanabilmesi için veri ambarı tasarımının ve mimari olarak uygulanacak çözümün belirlenmiş olması gerekir. Veri ambarı projesinde çalışacak insan kaynağının geliştirimde kullanılacak ürün ve teknolojiler konusunda bilgi sahibi olması geliştirimin başarılı bir şekilde tamamlanması için önemlidir. Veri ambarı geliştirimi genel olarak aşağıdaki aşamalardan oluşur.

- Veri Taşıma (Data Extract)
- Veri Dönüştürme (Data Transformation)
- Veri Yükleme (Data Load)
- Veri Sunum Katmanı Geliştirimi
- İş Zekası Uygulamaları ile Entegrasyonu

Veri ambarı ortamında bulunan veri taşıma, veri dönüştürme ve veri yükleme (ETL) sistemleri operasyonel sistemler ile veri sunum katmanı arasında bulunan tüm sistemleri kapsar [6]. ETL geliştirimi tamamlanan bir veri ambarında sonraki adım veri

sunum katmanında bulunacak data mart ve diğer veri tabanı nesnelерinin oluşturulması ve veri sunum katmanı ile iş zekası uygulamaları arasında entegrasyonun sağlanmasıdır.

3.4.1 Veri Taşıma (Extract)

Veri analizi aşamasında ihtiyaç duyulan operasyonel sistem tabloları ve diğer veri kaynakları belirlenir. Veri ambarının bulunduğu fiziksel veri tabanına ihtiyaç duyulan kaynak sistem verilerinin taşınması bu aşamada gerçekleşir. Farklı sistem ve ortamlarda bulunan verinin aynı ortama taşınması ile veri entegrasyonu sağlanmış olur. Veri ambarı projesinin başarılı olması için etkin bir veri taşıma uygulaması gereklidir. Bu nedenle, veri ambarına taşınacak her bir veri kaynağı için veri taşıma stratejisi geliştirilmelidir. Geliştirilecek strateji için aşağıdaki konulara dikkat edilmelidir [9].

- Kaynak sistemler ve veri yapıları
- Veri taşıma yöntemi (Tam ya da artışı)
- Veri taşıma sıklığı (günlük, aylık)
- Veri taşıma zamanı
- Veri taşıma sırası
- Hata düzeltme yöntemi

Veri taşıma yöntemi olarak mümkün olan durumlarda artışı yöntem kullanılması tercih edilir. Bu yöntemin kullanılabilmesi için kaynak sistem tablolarında verilerin değişiklik tarihinin bulunması gerekir. Bu şekilde sadece son veri taşıma tarihinden sonra değişen kayıtlar tespit edilir ve daha küçük bir veri hacmi ile taşıma yapılacağı için işlem daha hızlı tamamlanır. Ayrıca, veri taşımanın bir uygulama üzerinden mi yoksa manuel olarak mı yapılacağı belirlenmelidir.

Veri taşıma zamanının operasyonel sistemler üzerinde performans sorunu oluşturmayacak şekilde belirlenmesi gerekir. Ayrıca, operasyonel sistemlerin kendi akışlarında bulunan bazı işlerin beklenmesi de gerekebilir. Finansal sistemlerde bulunan gün sonu – gün başı işlemleri bu duruma örnek olarak gösterilebilir.

Veri taşıma sırası operasyonel sistemlerin veri okumaya uygun oldukları zamana göre planlanır. Ayrıca, birbiri ile ilişkili olan sistemlerden tutarlı veri alınması için veri taşıma sırasına dikkat edilir.

3.4.2 Veri Dönüştürme (Transformation)

Veri taşımanın tamamlanması ile operasyonel sistemlerde bulunan ham verinin veri ambarı ortamına alınması sağlanır. Veri ambarına yükleme yapmadan önce verinin temizlenmesi, veri kalitesinin artırılması ve veri ambarına yüklenecek bilgilerin üretilmesi gerekir. Veri üzerinde gerekli olan iyileştirme ve geliştirme işlemleri veri dönüştürme aşamasında yapılır. Veri temizleme ile kaynak sistemlerden eksik ya da hatalı gelen verilerin düzeltilmesi hedeflenir. Örneğin, kaynak sistemden gelen doğum tarihi bilgisi, içinde bulunulan günden daha ileri bir tarihsel bu tarih değeri hatalıdır ve veri ambarına yüklenmemelidir. Kaynak sistemden gelen bir tabloda mükerrer bulunan satırlar da yine kirli veriye örnektir ve veri ambarına yüklenmeden önce tekilleştirilmesi gerekir. Veri üzerinde gerekli iyileştirme ve geliştirme yapmak için kullanılan ana işlemler aşağıda belirtilmiştir [9].

- **Veri Seçimi:** Mükerrer kayıtların tekilleştirmesini ya da belirli şartları sağlayan veriye ulaşılmasını sağlar.
- **Veri Birleştirme:** Farklı sistemlerden gelen ilişkili verinin birleştirilmesini sağlar.
- **Veri Değiştirme:** Farklı sistemlerden gelen benzer verilerin standart hale getirilmesi, verilerin kullanılabilir ve anlaşılabilir olması için veri değiştirme yapılır.
- **Veri Özetleme:** Kaynak sistemden gelen alt seviyedeki verilerden daha genel bilgilerin üretilmesi sağlanır. Satış işlem verisi kullanılarak ürün ve mağaza seviyesinde özet satış verisinin oluşturulması veri özetlemeye örnek gösterilebilir.
- **Veri Zenginleştirme:** Kaynak sistemden gelen bir ya da daha fazla veriyi kullanarak daha kullanışlı bilgi üretilmesi sağlanır.

Yukarıda belirtilen veri dönüştürme işlemlerinin bir ya da birden fazlası kullanılarak sıklıkla yapılan veri dönüştürme tipleri aşağıda belirtilmiştir.

- Format Değişikliği
- Yeni Kod Üretme
- Hesaplanan ve Türetilen Değerler
- Veri Ayırıştırma
- Veri Birleştirme
- Veri Karakter Seti Dönüştürme
- Veri Ölçü Birimi Dönüştürme
- Tarih – Zaman Dönüştürme
- Veri Özetleme
- Veri Anahtarı Oluşturma
- Veri Tekilleştirme

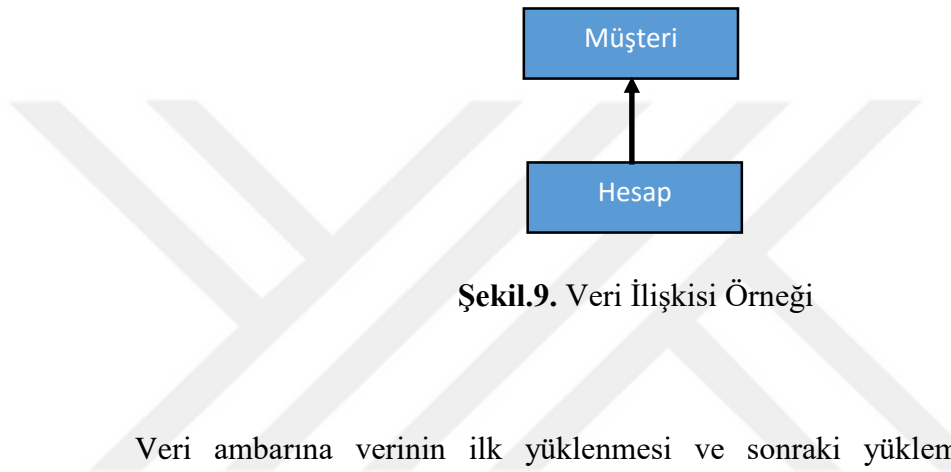
Farklı kaynaklardan gelen verinin dönüştürülmesi için gerekli olan zaman ve kaynağın önceden doğru olarak belirlenmesi mümkün değildir. Çünkü sistemler arasında ne tür uyumsuzluklar olduğu ve neler yapılması gerektiği ancak veri dönüştürme aşamasında görülür. Bu nedenle, proje planında olası uyumsuzlukları göz önünde bulunduran bir planlama yapılmalıdır. Aksi takdirde proje planında beklenmeyen ötelenmeler gerçekleşir.

Veri dönüştürme işlemleri için geliştirilmiş özel ürünler bulunmaktadır. Bu ürünler ile bir takım veri kalitesi analizleri ve veri dönüştürme işlemleri uygulanmaktadır. Veri dönüştürme ürünleri veri ambarı projelerine katkıda bulunsa da getirdiği satın alma ve bakım maliyetleri nedeni ile şirketler kendi çözümlerini de sıklıkla tercih etmektedir.

3.4.3 Veri Yükleme (Load)

Kaynak sistemlerden verinin taşınması ve sonrasında verinin dönüştürülmesi ile veri ambarına yüklenecek veri hazırlanmış olur. Hazırlanan verilerin ne zaman ve hangi sırada yükleneceğinin belirlenmesi gerekir. Bunun için verilerin hazır olma zamanı ve yüklenecek verilerin birbirlerine olan bağılıklarının belirlenmesi gerekir. Verilerin birbirlerine olan bağımlılığını tespit edebilmek için veriler arasındaki üst ast (parent child) ilişkisini gösteren veri modeli dikkate alınmalıdır. Veri bütünlüğünün sağlanması için üst ast ilişkisine göre yükleme yapılmalıdır. Örneğin, Şekil.9. ile

gösterilen veri modeline göre müşteri ve hesap sınıfları arasında bir üst ast ilişkisi bulunur. Veri modeline göre hesabın ilişkili olduğu bir müşteri bulunmaktadır. Bu nedenle, hesap sınıfına ait bir kaydın üzerindeki müşteri bilgisi ayrıca müşteri sınıfında bir kayıt olarak bulunmalıdır. Bunu sağlamak için her iki sınıfa yüklenecek kayıtlar aynı zamana ait olmalı ve öncelikle müşteri kayıtları yüklenmeli sonrasında hesap kayıtları yüklenmelidir. Bu şekilde hesap kayıtları üzerinde bulunan müşteri bilgisi ile müşteri tablosunda bulunan veriler karşılaştırılarak yükleme yapılır. Hesap üzerinde bulunan müşteri bilgisi müşteri tablosunda bulunmuyorsa veri ambarına yüklenmesi engellenir. Böylelikle veri ambarında veri bütünlüğü sağlanır.



Şekil.9. Veri İlişkisi Örneği

Veri ambarına verinin ilk yüklenmesi ve sonraki yüklemeler arasında farklılıklar bulunur. Veri yükleme sürecinin tamamı aşağıdaki aşamalardan oluşur [9].

- İlk Yükleme: Veri ambarının ilk defa verinin yüklendiği aşamadır.
- Düzenli Yükleme: Veri ambarı güncelleme sıklığına göre veride olan değişikliklerinin veri ambarına yansıtıldığı aşamadır. Değişiklik olmayan veriler tekrar yüklenmez.
- Tam Yükleme: İlk yüklemeye benzer şekilde bazı tabloların yeniden yüklendiği aşamadır. İlk yüklemeden farkı tüm tabloların yeniden tam yüklenmesi zorunluluğu bulunmamaktadır.

Veri ambarındaki tabloların ilk defa doldurulması zaman alan bir işlemdir. Ayrıca, ilk yükleme işlemi sırasında karşılaşılabilecek sorunlara karşı hazırlıklı olunmalıdır. Veri ambarına yapılacak ilk yükleme sırasında veri tipi, veri formatı gibi sebeplerle yüklenemeyen kayıtlar olabilir. Bu tür kayıtların nasıl takip edileceği, nasıl ve ne zaman düzeltileceği önceden belirlenmiş olmalıdır.

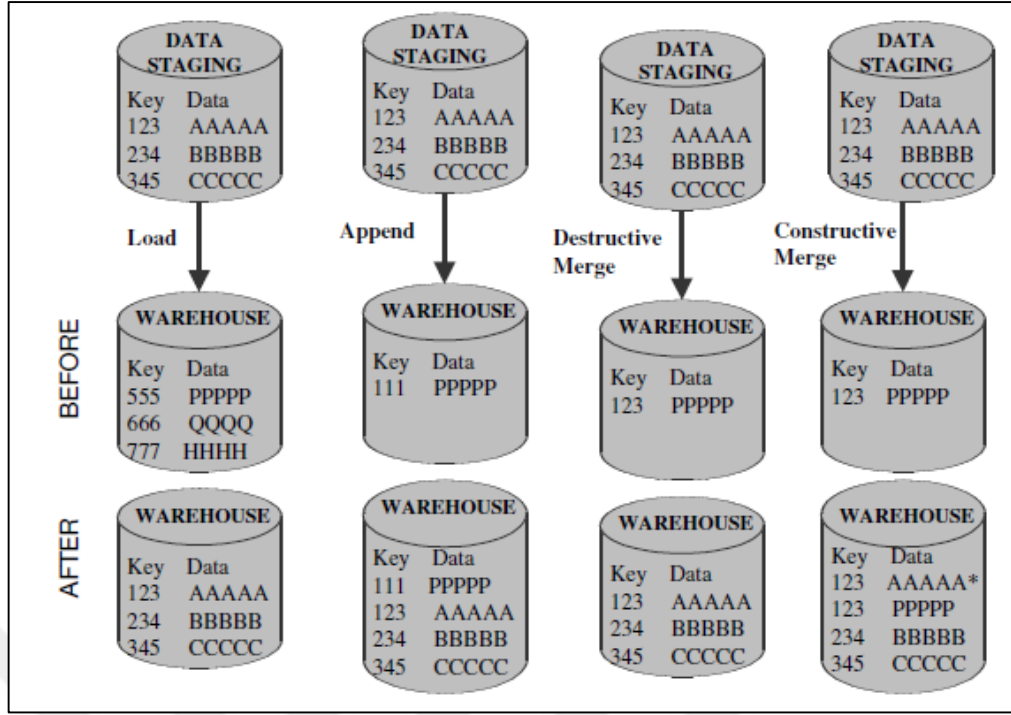
İlk yükleme sonrasında düzenli aralıklarla veri ambarında bulunan veri güncellenir. İlk yüklemedeki gibi uzun yükleme sürelerinin önüne geçilmesini sağlamak için düzenli yüklemelerde sadece yeni kayıtların ya da veri ambarında bulunup değişen kayıtların yüklenmesi sağlanır. Böylelikle daha küçük hacimde veri yüklenmesi sağlanır ve yükleme süresi kısaltılır. Kaynak sistemlerden gelen yeni veriyi ve değişen bilgileri tespit edebilmeye yönelik kod yazılması gerekir. Veri ambarında kullanılan ETL uygulamaları da yeni ve değişen bilgiyi tespit etmede kullanılır.

Düzenli aralıklarla yapılan artışı yüklemelerde zaman zaman sorunlar olabilmektedir. Kaynak sistemlerden gelen yeni verinin ya da değişen verinin tespit edilmesinde hatalar olabilir. Bu tür durumlarda veri ambarında bulunan verinin güncelliği bozulur. Veri ambarında bulunan verinin güvenilir ve doğru veri olması önemlidir. Bu nedenle, belirli dönemlerde veri ambarındaki bazı tabloların ilk yüklemedekine benzer şekilde tam yüklemesi yapılır.

Veri ambarına iş ihtiyaçlarına göre farklı şekillerde veri yüklemesi yapılır. Değişen verinin saklanma ihtiyacına göre uygun şekilde yükleme yapılır. İş ihtiyacına göre aşağıdaki dört yöntemden biri ile veri yüklemesi yapılır [9].

1. Mevcut veri silinir ve veri ambarında sadece son yüklemeyle ait kayıt bulunur.
2. Mevcut veri silinmez ve son yükleme ile gelen kayıtlar da veri ambarına eklenir. Veri ambarında bulunan bir kayıt gelirse ihtiyaca göre yüklenmesi engellenen veya izin verilen bir tasarım bulunur.
3. Mevcut veri silinmez ve son yükleme ile gelen kayıtlar veri ambarına eklenir. Yüklenen tabloda anahtar (key) bilgileri eşleşen kayıt varsa ilgili mevcut kayıtlar güncellenir.
4. Mevcut veri silinmez ve son yükleme ile gelen kayıtlar veri ambarına eklenir. Yüklenen tabloda anahtar (key) bilgileri eşleşen kayıt varsa ilgili mevcut kayıt değiştirilmeden korunur, yeni kayıt eklenir ve güncel kayıt olarak işaretlenir.

Veri yükleme yöntemleri görsel olarak Şekil.10. ile gösterilmiştir.



Şekil.10. Veri Yükleme Yöntemleri

Kaynak: Ponniah, P. (2001), *Data Warehouse Fundamentals / A Comprehensive Guide for IT Professionals*, s. 281, Wiley Publications, USA.

3.4.4 Veri Sunum Katmanı Geliştirimi

Veri ambarına yüklenen veriye farklı erişim yöntemleri ile ulaşılabilir. Veriye erişim doğrudan veri ambarına erişerek olabileceği gibi, veri sunum katmanında oluşturulan tablolar üzerinden de erişilebilir. Veri sunum katmanı kullanıcı ihtiyaçlarına ve mimari tasarıma göre geliştirilir. Veri sunum katmanı ayrı bir fiziksel ortam olabileceği gibi veri ambarı ile aynı fiziksel ortamda da oluşturulabilir. Veri ambarı tasarım yöntemine göre veri kurumsal veri modeline göre oluşturulmuş fiziksel tablolar (Inmon yaklaşımı) ya da data martlar (Kimball yaklaşımı) içerisinde depolanır. Veri modeline göre oluşturulan tablolar üzerinden veriye ulaşmak için birçok tablodaki verinin birleştirilmesi gerekir. Kullanıcılar tarafından bu şekilde sorguların oluşturulması teknik zorluklar içerir. Bu nedenle veriye daha kolay, daha hızlı ve sadece kendileri ile ilgili veriye ulaşabilmeleri için data martlar oluşturulur. Data martlardan oluşan bir veri ambarında da birden fazla data mart kullanılarak yeni data martlar üretilebilir.

Veri ambarı projesinin yapılış amacının gerçekleştiği aşama veri sunum katmanının oluşturulmasıdır. Çünkü son kullanıcılar bundan önceki aşamalarda üretilmiş verilerin hiçbirine erişemez ve sadece veri sunum katmanındaki veriyi kullanır. Veri sunum katmanına kadar çok iyi tasarlanmış ve geliştirilmiş bir veri ambarı, veri sunum katmanının yetersiz olması durumunda başarısız olarak değerlendirilir.

Kullanıcılar çok farklı amaçlar için veriye erişir. Veriye erişmek için de amacına uygun uygulamalar kullanılır. Her bir kullanıcı ihtiyacı ve veri erişim uygulaması için farklı veri sunum nesnelere oluşturulması gerekebilir. Bu nedenle, veri sunum katmanının oluşturulması için iş ve teknik ihtiyaçların detaylı olarak analiz edilmesi gerekir. Veri sunum katmanında bulunan nesnelere manuel olarak oluşturulabildiği gibi bazı uygulamalar vasıtasıyla da oluşturulur.

3.4.5 İş Zekası ve Veri Analiz Uygulamaları

Veri sunum katmanında oluşturulan veri nesnelere iş zekası ve analiz uygulamaları üzerinden erişilir. Bazı uygulamalar doğrudan veri sunum katmanına erişerek veriyi gösterir ve bu veri üzerinden analizler yapılmasını sağlar. Standart raporlama uygulamaları bu şekilde çalışır. Bazı uygulamalar ise veri sunum katmanındaki veriyi kullanarak kendi veri tabanlarını ve tablolarını oluşturur. Veri madenciliği ve veri analitiği uygulamaları bu şekilde çalışır. Bu nedenle, iş zekası ve veri analizi amacıyla kullanılan uygulamaların teknik altyapısının hazırlanması, kurulumu ve veri ambarı ile entegrasyonun sağlanması gerekir. Ayrıca, bu uygulamaların veriyi etkin bir şekilde okuması ve işlemesi için gerekli optimizasyon çalışmaları yapılır.

3.5 Veri Ambarı Testleri

Veri ambarı geliştiriminde kaynak sistemlerden verinin alınmasından veri sunum katmanının oluşturulmasına kadar geçen tüm aşamalarda fonksiyonel ve teknik test ihtiyaçları bulunur. ETL ve veri sunum katmanı geliştirmeleri ile birlikte iş zekası uygulamalarının entegrasyon testleri yapılmalıdır. Ayrıca, veri ambarına yapılacak ilk yükleme ve sonrasında yapılacak düzenli yüklemeler arasında farklılıklar olduğu için

veri doğruluk testleri her iki yükleme tipi için de yapılmalıdır. Veri ambarı projelerinde ETL geliştirim testleri, veri yükleme testleri, kullanıcı kabul testleri ve veri ambarı bakım testlerinin yapılması gerekir [19]. Testlere başlamak için tüm veri ambarı geliştiriminin tamamlanması gerekmez. Veri ambarı geliştirim tamamlandıkça testlere başlanır. Testlerin gerçekleştirilmesi için üretim ortamına benzer veri ortamları oluşturulur.

- 1. ETL Geliştirim Testleri:** Kaynak sistemden verinin eksiksiz ve değişmeden alındığı test edilir. Veri analizi sırasında belirlenen veri ihtiyaçlarının veri dönüştürme ve birleştirme işlemleri ile doğru olarak üretildiği test edilir. Ayrıca, kaynak sistemden veri taşıma ve veri dönüştürme işlemleri için performans testleri gerçekleştirilir. Bu testler yazılım ekipleri tarafından gerçekleştirilir.
- 2. İlk Yükleme Testleri:** İlk yükleme için hazırlanan verinin eksiksiz ve değişmeden veri ambarına yüklendiği test edilir. Veri sunum katmanında oluşturulan veri nesnelere veri ambarı ile tutarlı olduğu kontrol edilir. Veri yükleme performans testleri gerçekleştirilir. Bu testler yazılım ekipleri tarafından gerçekleştirilir.
- 3. Kullanıcı Kabul Testleri:** Veri ambarı geliştirimi uçtan uca tüm fonksiyonları ile test edilir. İş ihtiyaç analizi sonucunda belirlenen veri ihtiyaçlarının eksiksiz ve doğru olarak karşılandığı kontrol edilir. Veriye erişmek için kullanılacak iş zekası uygulamalarının entegrasyonu ve performansı test edilir. Bu testler son kullanıcılar tarafından yapılır. Yazılım ekipleri gerekli test ortamının ve test verisinin sağlanması için destek olur, ayrıca, iletilen bulguları inceler ve gerekli ise düzeltmelerini yapar.
- 4. Düzenli Yükleme Testleri:** İlk yükleme sonrasında yapılan düzenli yüklemelerde değişen ve yeni gelen kayıtların düzgün olarak yüklendiği kontrol edilir.

3.6 Üretim Ortamına Geçiş

Veri ambarı üretim ortamlarının hazırlanması, uygulama entegrasyonlarının sağlanması ve veri ambarı testlerinin tamamlanması sonrasında veri ambarı

geliřtirimleri üretim ortamına taşınır. Ayrıca, veri ambarı üretim ortamında kullanıcı tanımları ve yetkilendirmeleri yapılarak üretim ortamı kullanıma hazır hale getirilir.



4. DİNAMİK VERİ AMBARI TASARIMI

Farklı sistemlerde bulunan verilerin birleştirilerek ihtiyaç duyulan bilgilerin üretilmesi için geleneksel veri ambarı çözümleri uzun yıllardır uygulanmaktadır. Teknolojinin hızlı bir şekilde gelişmesi ve yaygınlaşması sonucunda ilgilenilmesi gereken veri hacmi, veri kaynağı ve veri tipleri artmıştır. Ayrıca, rekabet avantajında veriden sağlanan faydanın etkisi de giderek artmaktadır. Bu nedenle, veri ambarı tasarımları hızlı bir şekilde doğru, kaliteli ve yeterli seviyede bilgi sunmalı, değişen ve gelişen piyasa koşullarına göre ihtiyaç duyulan veriyi dinamik olarak sağlamalıdır.

Yapılan araştırmalar veri ambarı projelerinin yüksek oranda başarısızlıkla sonuçlandığını göstermektedir [1]. Geleneksel veri ambarı projelerinin başarısız olarak değerlendirilmesinin birçok sebebi bulunur. Proje hedeflenen zaman içerisinde tamamlanmamış, proje bütçesi aşılmış, veri ambarı iş ihtiyaçlarını karşılamıyor ya da veri ambarının sistem performansı yetersiz olarak değerlendirilebilir. Dinamik veri ambarı yaklaşımı ile geleneksel yaklaşımda başarısızlık değerlendirmelerine sebep olan etkenlerin düzeltilmesi hedeflenir. Bu nedenle, veri ambarı projelerinin başarısız olarak değerlendirilmesine sebep olan etkenlerin ortadan kaldırılarak yeni bir veri ambarı tasarım yaklaşımı geliştirilmesine ihtiyaç duyulmuş ve sonuç olarak dinamik veri ambarı (Dynamic Data Warehousing - DDW) yaklaşımı ortaya çıkmıştır.

IBM'e göre dinamik veri ambarı bir ürün, uygulama ya da bir çözüm paketi değildir. Dinamik veri ambarı, farklı tiplerdeki bilgilerin birleştirilerek, dönüştürülerek, işlenerek ve analiz edilerek daha dinamik bir iş anlayışının ortaya konmasını sağlayan bir yaklaşımdır [20]. Dinamik veri ambarının amacı iş ihtiyaçlarına çok hızlı cevap verebilen bir yaklaşım üretmektedir. İş ihtiyaçları zaman içerisinde büyüyen, gelişen ve evrim geçiren bir yapıya sahiptir. Veri ambarı da bu değişime ayak uydurmalıdır. Dinamik veri ambarı yaklaşımın en öncelikli hedefi hızlı bir şekilde kaliteli bilgi sunarak iş ihtiyaçlarının karşılanmasıdır. Bunu sağlamak için dinamik veri ambarı yaklaşımı ile mümkün olan aşamalarda manuel kod yazım teknikleri yerine veri ambarı otomasyon yazılımları ile çözüm üretilir.

Veri ambarının geleceği veri ambarı otomasyonudur. Veri ambarı otomasyonu ile iş ihtiyaçları geliştirimin ileri aşamalarında dahi değiştirilebilir ve genel olarak bu değişiklik o ana kadar yapılmış çalışmaların daha az bozulmasına ve kullanılamaz

olmamasını sağlar. Bu verimlilik zamandan, insan kaynağından ve maliyetten kazanç elde edilmesini sağlar. Veri ambarı otomasyonu geliştirim standartlarını zorlaması ve geliştirimi standart hale getirmesi nedeniyle daha kaliteli bir geliştirim yapılmasına katkıda bulunur [21].

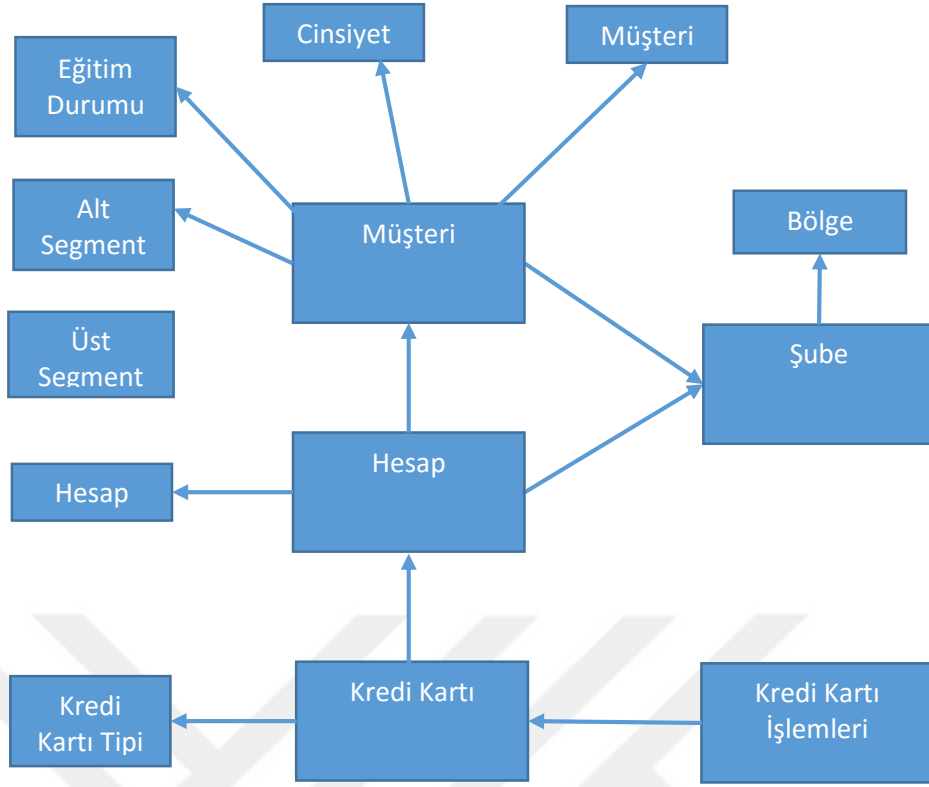
4.1 Geleneksel Yaklaşımda Karşılaşılan Sorunlar

4.1.1 Veri Modelleme

İş ihtiyaçlarının analiz edilmesi ile birlikte veri ihtiyaçları belirlenir. Veri ihtiyaçlarının belirlenmesi ile veri ambarı üzerinden gerçekleştirilecek iş fonksiyonları için gerekli olan veri listesi oluşur. Ortaya çıkan veri listesi veri sunum katmanında gösterilecek veriler olarak düşünülebilir.

İhtiyaç duyulan veri listesinin belirlenmesi sonrasında veri analizi yapılır. Veri analizi veri listesinde bulunan verilerin nasıl hesaplanacağı, hangi yapıda ve tipte saklanacağı gibi detaylar belirlenir. Veri analizi ile ortaya çıkan bilgilere göre veri modelleme yapılır.

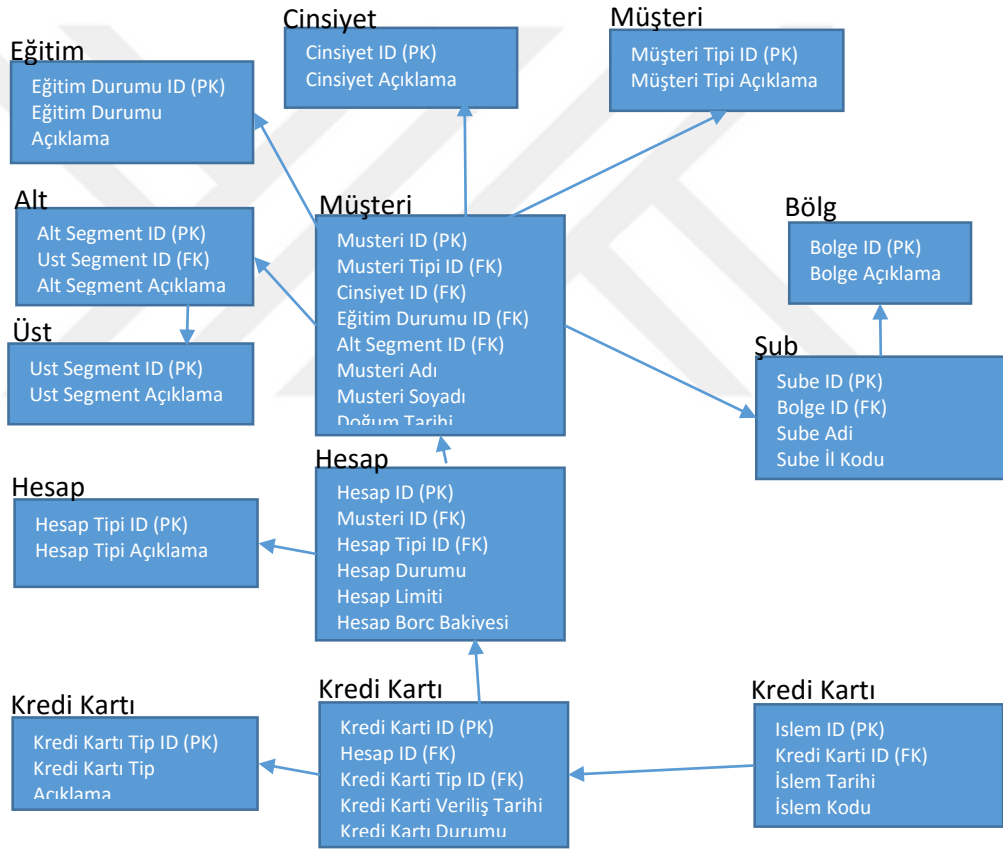
Veri modellemenin ilk aşaması kavramsal modellemedir. Bu aşamada ortaya çıkan veri modeli veri ile ilgili genel bir bilgi verir. Veri ambarındaki ana tablolar ve tablolar arasındaki ilişkiler gösterilir. Bu aşamada veri modelleme uzmanları ile birlikte veri hakkında bilgi sahibi olan operasyonel sistem kullanıcıları birlikte çalışır. Veri ambarına birçok farklı operasyonel sistem ve kaynaktan veri alındığı için ihtiyaç duyulan katılımcı sayısı yüksektir. Çünkü her operasyonel sistemin kullanıcısı sadece kendi sistemindeki veriden sorumludur ve kendi sisteminin verisini bilir. Bu nedenle kavramsal modelleme çalışmaları kalabalık bir katılımcı grubu ile yürütülür. Tüm sistemler arasında kullanılan ortak veriler olduğu için çalışmaların bir kısmı tüm katılımcılar birlikteyken yapılır. Kavramsal modelleme veri ambarının geliştirileceği teknolojik platformdan bağımsızdır. Kavramsal veri modeli ileriki modelleme aşamaları ile veri ambarının geliştirileceği teknolojik platforma göre özelleştirilir. Kavramsal modelleme için bazı modelleme uygulamaları kullanılabilir fakat bu bir zorunluluk değildir. Sadece kağıt kalem kullanılarak da kavramsal bir model oluşturulabilir. Şekil.11. de örnek bir kavramsal veri modeli gösterilmiştir.



Şekil.11. Kavramsal Model Örneği

Kavramsal veri modelinin oluşturulması sonrasında mantıksal veri modeli aşamasına geçilir. Mantıksal veri modeli kavramsal veri modelinin geliştirilmiş ve detaylandırılmış halidir. Bu aşamada veri modelleme konusunda uzman kişiler çalışır. Teknik uzmanlık ve zaman gerektiren geliştirim sürecinden dolayı veri hakkında bilgi sahibi olan operasyonel sistem kullanıcıları bu aşamada çok fazla yer almaz. İhtiyaç duyulan durumlarda kullanıcılardan bilgi alınabilse de doğrudan sorumlu olmadıkları için kullanıcılar bu aşamada geliştirimden kopmaya başlar. Veri modeli geliştirilirken zaman zaman kullanıcılar ile değerlendirmeler yapılır. Fakat modeli anlamak için teknik bilgi gerekmesi ve doğrudan sorumlu olunmaması nedeni ile yanlış ya da eksik modellemeler gözden kaçmaktadır. Bu aşamada veri listesinde bulunan her bir veri için modelleme yapılır. Veri ambarının oluşturulacağı fiziksel veri tabanı nesnelinde ihtiyaç duyulabilecek tüm bilgiler modelde gösterilir. Tablo ve kolon isimleri, primary ve foreign key gibi teknik bilgiler modelde yer alır. Mantıksal veri modeli veri ambarının geliştirileceği teknolojik platformdan bağımsız olarak oluşturulur. Fiziksel modelleme aşamasında veri ambarının geliştirileceği teknolojik platforma göre özelleştirilir. Mantıksal modelleme sıklıkla bu amaç için geliştirilmiş modelleme

uygulamaları üzerinde geliştirilir. Veri modelleme uygulamaları ile mantıksal model geliştirildikten sonra fiziksel modelleme aşamasında kullanılacak çıktılar üretilebilir. Veri ambarı geliştiriminin yapılacağı teknolojik platforma göre veri tabanı kodları bu uygulamalar tarafından sağlanabilir. Fakat veri modelleme uygulamalarının kullanılması bir zorunluluk değildir. Microsoft office ve benzeri uygulamalar kullanılarak da mantıksal bir model oluşturulabilir. Bu durumda fiziksel modelleme aşamasında ihtiyaç duyulan tablo yaratma kodlarının manuel oluşturulması gerekir. Daha kaliteli ve hızlı bir şekilde kodları oluşturmak için mümkün olan durumlarda veri modelleme uygulamalarının kullanılması tercih edilir. Şekil.12. ile örnek bir mantıksal veri modeli gösterilmiştir.



Şekil.12. Fiziksel Model Örneği

Mantıksal veri modeli ile veri listesinde yer alan tüm alanlar için fiziksel veri tabanının oluşturulmasında ihtiyaç duyulan tüm bilgiler belirlenmiş olur. Veri modelleme uygulaması kullanılması durumunda fiziksel veri tabanının oluşturulmasında kullanılacak kodlar üretilebilir. Veri modelleme uygulamasının

kodları üretmemesi ya da modelleme uygulaması kullanılmaması durumunda kodların manuel oluşturulması gerekir. Kodların manuel üretilmesi hataya açık bir durumdur. Veri tabanı nesnelere özelliği olması gerekenden farklı bir şekilde oluşturulursa ancak veri yükleme aşamasında karşılaşılabilecek sorunlar ile fark edilebilir. Fakat veri yükleme aşaması bu tür sorunların tespit edilmesi için oldukça geç bir aşamadır. Ayrıca, fiziksel veri ambarının oluşturulması için veri tabanı yaratma, tablo yaratma, tablolar arası ilişkiler, kullanıcı yetkilendirme gibi birçok farklı tipte kod oluşturulması gerekir. Bu nedenle mümkün olduğu sürece insan faktörünü ortadan kaldıracak şekilde veri otomasyon uygulamaları kullanılarak geliştirim yapılmalıdır.

Sonuç olarak; geleneksel veri modelleme geliştirim aşamalarında aşağıdaki sorunlar bulunmaktadır ve dinamik veri ambarı tasarım yaklaşımı ile bu sorunların giderilmesi hedeflenir.

- 1. Veri modellemenin üç farklı aşamada yapılması:** Farklı kullanıcı grupları ile farklı nitelikte modeller oluşturulur. Genelden detaya doğru giden bir yaklaşım benimsenir. Fakat detay aşamalarda ilerledikçe teknik ve son kullanıcılar arasında kopmalar başlar. Bu nedenle detay aşamalarda yapılan modellemenin genel olarak doğru olup olmadığı son kullanıcılar tarafından takip edilememekte ve yapılan hatalar çok ileriki aşamalarda ortaya çıkmaktadır.
- 2. Mantıksal ve fiziksel veri modeli anlamak için teknik bilgi gerekmesi:** Kavramsal modelleme sonrasında oluşturulan modeller son kullanıcıların görüşüne sunulsa da teknik bilgi gerektirmesi nedeniyle model son kullanıcılar tarafından tam olarak anlaşılmamaktadır. Bu nedenle modelde yapılan hatalar zamanında fark edilememektedir.
- 3. Eksik ya da hatalı modellemenin etkisi:** Modellemenin herhangi bir aşamasında eksik ya da hatalı bir işlem yapılması durumunda en başa dönüp tüm modellerin güncellenmesi gerekir. Örneğin, veri listesinde bulunan bir verinin unutulması durumunda önce kavramsal model üzerindeki etkisi değerlendirilir ve gerekli ise ekleme yapılır. Kavramsal modelde detay bilgileri olmadığı için bu kısmı etkilemeyebilir. Sonrasında unutilanan verinin mantıksal modelde nereye eklenmesi gerektiği analiz edilir. Bu aşamada son kullanıcılardan ya da kaynak sistem uzmanlarından destek alınması gerekebilir. Son olarak,

değişikliğin fiziksel veri ambarına da yansıtılması gerekir. Manuel olarak kod yazılan bir ortamda ufak bir değişikliğin maliyeti oldukça yüksek olabilir.

- 4. Manuel kod yazımlarında hata riski olması:** Modelleme doğru yapılmış olsa bile manuel kod yazımı sırasında istenmeyen hataların oluşmasına sebep olabilir. Bu tür hatalar projenin ileriki aşamalarında tespit edilebilir. Bu nedenle mümkün olduğu sürece insan faktörünün sebep olabileceği hataların önleyen çözümler tercih edilmelidir.
- 5. Model bakım maliyeti:** Veri ambarı geliştiriminin tamamlanması sonrası gelen yeni ihtiyaçlar için tüm modellerin güncellenmesi gerekir. Bu durum modeller için bir bakım maliyeti ve yeni taleplerin geliştirim süresini arttıran bir etki oluşturur.

4.1.2 Metadata Yönetimi

Veri ambarı ortamında bulunan önemli bileşenlerden biri de metadadır. Metadata, genel tanımı ile veri hakkında veri, yazılım uygulamaları ve verinin ortaya çıkışından bu yana bilgi işleme süreçlerinin bir parçası olmuştur. Fakat veri ambarı dünyasında metadatanın önemi daha da artmıştır. Çünkü veri ambarının en etkin şekilde kullanılmasını sağlar [3]. Metadatanın veri ambarı ortamında önemli olmasını sağlayan özellikleri aşağıda belirtilmiştir [9].

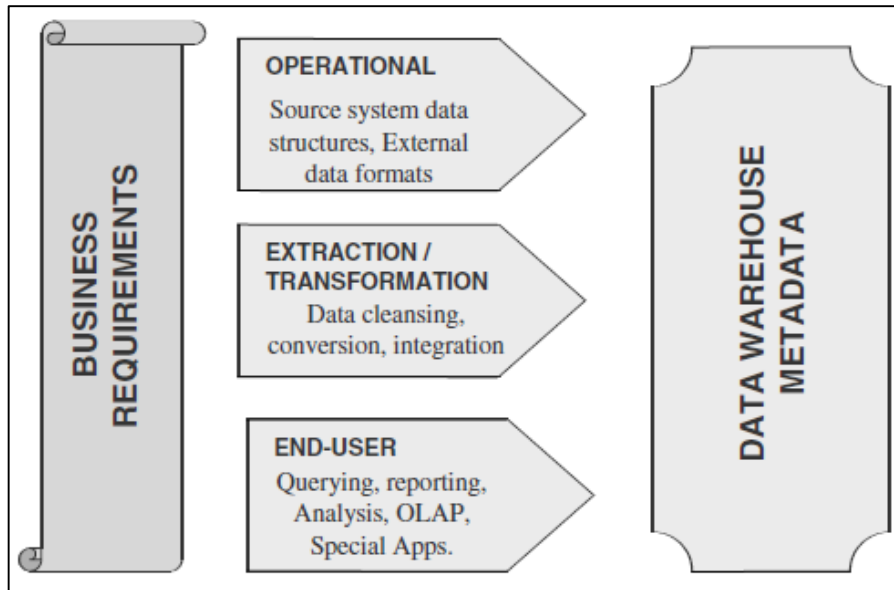
- Veri ambarının tüm parçalarını birleştiren bilgi verir.
- Yazılım geliştiricilerine ve analistlere veri ambarı içeriği ve yapısı ile ilgili teknik bilgiler sağlar.
- Son kullanıcılar için veri ambarında bulunan verinin ulaşılabilir ve anlaşılabilir olmasını sağlar.

Metadata veri ambarında bulunan veri ve verinin oluşturulma süreci ile ilgili bilgi veren bir kaynak olarak düşünülebilir. Veri ambarında bulunan verinin tanımı, tipi, kaynağı, nasıl hesaplandığı gibi bilgiler metadada bulunur. Metadata veri ambarının içerdiği veriyi ve veri ambarı işleyişini tarif eden bir katalog görevi görür. Metadata geliştirimi veri ambarı projelerini diğer yazılım projelerinden farklılaştıran unsurlardan biridir [9].

Veri ambarında bulunan metadata 3 ana kategoriye ayrılır [9].

- **Operasyonel:** Veri ambarına birçok farklı kaynak sistemden veri aktarılır. Kaynak sistemlerden farklı formatta ve tipte veriler bulunur. Veri ambarına yüklenen veriler için kaynak sistemdeki bu verilerin birleştirilmesi gerekir. Son kullanıcılara sunulan verinin üretilmesinde kullanılan kaynak sistemler ve tabloların bilinmesi gerekir. Operasyonel veri kaynakları ile ilgili tüm bilgiler operasyonel metadata'da bulunur. Örnek: Kaynak sistemler, tablo şemaları
- **Veri Taşıma ve Dönüştürme:** Veri taşıma işlemlerinde kullanılan kaynak tablolar, veri taşıma sıklığı, veri taşıma yöntemi ve kuralları ile birlikte veri dönüştürme aşamasında yapılan işlemlere ait bilgiler bu kategori içerisinde bulunur. Örnek: Veri taşıma kuralları, veri hesaplama kuralları, veri modeli
- **Son Kullanıcı:** Son kullanıcılara ihtiyaç duyduğu veriye nasıl ulaşacağını gösteren bilgiler bu kategori içerisinde bulunur. Örnek: Veri tanımları, Veri ambarı tabloları ve kolonları, Hazır raporlar,

Metadata gerekliliği iş ihtiyaçlarına verimli bir şekilde çözüm sağlanması amacıyla ortaya çıkmıştır. İş ihtiyaçlarının metadata üzerindeki etkisi Şekil.13. ile gösterilmiştir [9].



Şekil.13. İş İhtiyaçlarının Metadata Etkisi

Metadata içerdiği teknik ve fonksiyonel bilgiler nedeniyle hem yazılım ekiplerinin hem de son kullanıcı ekiplerin faydalandığı bir kaynaktır. Metadatanın olmadığı bir veri ambarında son kullanıcıların ihtiyaç duyduğu bir veriye ulaşmaları uzun zaman alır. Ayrıca, çaba harcayarak zorlukla ulaştıkları kaynak doğru olmayabilir. Bu tür durumlarda son kullanıcılar yazılım ekiplerinden destek almak zorunda kalır. Bu nedenle, metadata eksikliği ya da yetersizliği son kullanıcılar tarafında yazılım ekiplerine bağımlılık oluşturur. Yazılım ekiplerine olan bağımlılık son kullanıcıların veriden etkin ve hızlı bir şekilde faydalanmasını engeller. Metadata gerekliliği yazılım ekipleri için de geçerlidir. Metadatanın eksik ya da yetersiz olması durumunda son kullanıcılara verilmesi gereken destek nedeniyle iş verimsizliği oluşur. Veri ambarında çalışan işler, uygulanan yöntem ve kurallar metadadata bulunur. Metadatanın oluşturulması veri ambarı geliştiriminde kişi bağımlılığını da azaltan bir etkidir. Metadata bilginin kurum içerisinde sürekliliğini sağlayarak bilginin kaybolmasını önler.

Metadata tüm veri ambarı bileşenleri ile ilgili bilgi içerir. Bu özelliği ile birçok farklı bileşeni olan veri ambarının birleştiren yapısıdır. Fakat bu yapıyı oluşturmak için proje boyunca özel çaba gösterilmelidir.

Geleneksel yaklaşımda, her geliştirim süreci için metadata dokümanları manuel olarak oluşturulur ve sürdürülür. Geliştirimde kullanılan bazı uygulamalar kendi metadalarını üretse de metadata çıktılarının birleştirilmesi gerekir. Data Warehousing Institute tarafından yapılan araştırmada katılımcıların %86'sı metadata yönetim stratejisinin önemini anladıklarını belirtmiştir. Fakat sadece %9'u metadata geliştirimi yapmış, %16'sı ise metadata geliştirim çalışmalarının devam ettiğini belirtmiştir [9]. Kurumların çok büyük kısmının metadatanın önemini anlamış olmasına rağmen henüz metadata geliştirimine başlamadığı görülmektedir. Bunun sebebi metadata geliştiriminde endüstri standardının henüz tam olarak oluşmaması ve metadata geliştiriminde yer alan zorluklardır. Metadata geliştiriminde karşılaşılan ana zorluklar aşağıda belirtilmiştir [9].

- Veri ambarı geliştiriminde kullanılan her uygulama kendi metadata bilgisini üretir. Birkaç uygulama kullanıldığında elde edilen farklı formattaki metadata bilgilerinin birleştirilme zor bir süreçtir.
- Genel kabul gören endüstri standardı oluşmamıştır.

- Değişen metadata bilgilerinin versiyonlanarak saklanması zorlu bir işlemdir.
- Birçok farklı kaynak sistemden veri alınan büyük veri ambarları için veri kaynakları ile ilişkili metadata'nın birleştirilmesi devasa bir iştir. Veri kaynaklarında çelişen standartlar, formatlar, veri isimlendirme yöntemleri, veri tipleri, veri değerleri, iş kuralları ve veri ölçüm birimleri ile uğraşmak gerekir. Kaynak sistemlerde yetersiz olan veri kalitesinin telafi edilmesi gerekir.

Metadata geliştiriminin zaman gerektiren zorlu bir iş olması nedeni ile proje geliştirim süresinde zaman zaman önceliğini kaybeder. Öncelik veri ambarına verinin yüklenmesi ve veri ambarının kullanıma açılması olur. Metadata'nın geçiş sonrası tamamlanabileceği düşünülür. Fakat proje tamamlandıktan sonra projenin önemi ve yönetim tarafından verilen desteği azalır. Bunun sonucunda, çoğunlukla, proje kaynakları farklı işlere yönlendirilir ve metadata geliştirimi sonlandırılmaz.

4.1.3 Veri Sunum Katmanı ve İş Zekası Uygulamaları Entegrasyonu

Kullanıcıların veri ambarına yüklenen veriye erişebilmeleri için veri sunum katmanı oluşturulur. Veri sunum katmanı kullanıcı ihtiyaçları ve mimari tasarım göz önüne alınarak geliştirilir. Veri sunum katmanının kullanıcı ihtiyaçlarına cevap verebilecek şekilde tasarlanması oldukça önemlidir. Çünkü tüm veri ambarı geliştirim sürecinin kullanıcılara görünür olduğu yer veri sunum katmanıdır. Veri ambarı geliştirimi bir süreç olarak düşünüldüğünde, süreç girdilerinin belirli işlemlerden geçmesi sonucunda oluşan çıktı veri sunum katmanıdır. Kullanıcılar veri ambarının başarı değerlendirmesini veri sunum katmanı kullanım tecrübelerine göre yapar. Veri sunum katmanının başarılı olarak değerlendirilebilmesi için aşağıdaki niteliklere sahip olması beklenir.

- Gerekli ve yeterli seviyede bilgi içermeli
- Kaliteli ve doğru bilgi içermeli
- İhtiyaç duyulan zamanda hazır olmalı
- Kolay erişilebilir olmalı
- Veri sorgulama ve işleme performansı tatmin edici olmalı

- Değişen ve gelişen kullanıcı ihtiyaçlarına hızlı bir şekilde cevap vermeli

Veri ambarı geliştirimi birçok farklı sistemden verinin taşınması ve sonrasında verinin işlenerek iş ihtiyaçlarını karşılayacak bilgiye dönüştürülmesi ile gerçekleştirilir. Tüm bu işlemlerin yapılması için farklı uzmanlık alanlarına sahip insan kaynağı ve teknolojik altyapı gerekir. Veri ambarı projelerinin geliştirim süresi uzun ve geliştirim maliyeti yüksektir. Veri ambarı geliştirimine yatırım yapan kurumlar bir an evvel ve en etkin şekilde veri ambarından faydalanmak ister. Bu beklentinin karşılanması için kullanıcı ihtiyaçlarına cevap veren bir veri sunum katmanı tasarlanmalıdır.

Veri ambarı tek bir kullanıcı grubunun kullanması için tasarlanmaz. Veri ambarı içerdiği veri ile farklı kullanıcı gruplarının ihtiyacını karşılamalıdır. Veri ambarının genel kullanım amaçları ve ihtiyaç duyulan veri niteliği Tablo.4. ile gösterilmiştir.

Tablo.4. Veri Ambarı Kullanım Amacı ve Veri Niteliği

Kullanım Amacı	İhtiyaç Duyulan Veri Niteliği
Operasyonel Raporlama	Kullanıcı ihtiyaçlarına göre düzenli olarak alınması gereken raporlamalar bu kapsamdadır. İş zekası uygulamaları üzerinde oluşturulan hazır raporlar ile veriye erişilir. Hazır raporlar mevcut datamartlar üzerinden raporu oluşturur ya da raporlama konusuna özel hazırlanmış datamartlar kullanılır. Genellikle güncel veri raporlanır.
Veri Analizi	Analiz edilmek istenen konuyla ilgili detay bilgiler bir ya da daha fazla datamart ile kullanıcıya sunulur. Kullanıcılar iş zekası uygulamaları aracılığıyla ya da doğrudan veriye erişerek analiz yaparlar.
Veri Madenciliği / Analitik	Güncel veri ile birlikte mutlaka geçmiş dönemlere ait verilere ihtiyaç duyulur. Hem detay hem de özet veri kullanılır.
Karar Destek Sistemleri	Büyük resmi gösteren özet veriye ihtiyaç duyulur.

Farklı amaçlar için farklı veri kümeleri ile çalışacak kullanıcılara veri sunum katmanında çok sayıda veri tabanı nesnesi oluşturulması gerekir. Her bir veri tabanı nesnesinin geliştirilmesi sonrasında veriyi doğru gösterdiğinin test edilmesi de gerekir.

İş zekası uygulamaları ile veri sunum katmanına erişilerek operasyonel raporlar alınır ve veri analizleri gerçekleştirilir. Ayrıca, iş zekası uygulamalarının bir çoğunda dinamik rapor oluşturma fonksiyonları da bulunur. İş zekası uygulamaları kendi metadatalarını kullanarak kullanıcılara önyüzleri aracılığıyla kod yazmadan rapor oluşturma olanağı verir. Fakat bu fonksiyonları kullanabilmek için iş zekası uygulamasının kendi metadatasının oluşturulması gerekir. İş zekası uygulaması veri ambarındaki veri modelini kendi metadata formatında olmadığı için kullanamaz. Bu nedenle, veri ambarı için geliştirilen veri modelinin bir benzeri iş zekası uygulaması için de geliştirilmelidir. Birçok iş zekası uygulamasında metadata geliştirimi manuel yapılıdır. Büyük kurumlarda sıklıkla farklı ihtiyaçlar için farklı iş zekası uygulamaları kullanılır. Bu durum iş zekası metadata geliştiriminin uygulama adedi kadar yapılmasını gerektirir. Ayrıca, veri ambarına yapılacak tüm eklemelerin iş zekası uygulamalarına da yansıtılabilmesi için yeniden metadata geliştirimi gerektirir. Bu durum hem geliştirim sürecini uzatır hem de metadata yönetimini zorlaştırır.

Geleneksel veri ambarı geliştirim yaklaşımında veri sunum katmanı manuel geliştirilir. Geliştirim yapılması gereken nesne adedi arttıkça geliştirim süresi de aynı şekilde artar. Bu durum veri sunum katmanının oluşturulması için ihtiyaç duyulan süreyi uzatır. Ayrıca, metadata entegrasyonu veri ambarı teknolojisinde yeni ve gelişen bir konu olduğu için geleneksel yaklaşımda tam bir karşılığı bulunmamaktadır. Bu sebeplerden dolayı, geleneksel yaklaşımda, veri sunum katmanının hazırlanması ve iş zekası uygulamaları ile entegrasyonu zaman alan ve yönetimi zor bir süreçtir. Veri ambarı başarı değerlendirmesini doğrudan etkileyen bu sorunlara dinamik veri ambarı yaklaşımı ile çözüm getirilmesi hedeflenir.

4.2 Geleneksel Yaklaşımda Karşılaşılan Sorunlara Dinamik Yaklaşım Çözümleri

4.2.1 Veri Modelleme

Geleneksel yaklaşımda veri modelleme üç aşamada tamamlanır. Öncelikle ihtiyaç duyulan veriler kavramsal olarak modellenir. Sonrasında mantıksal modelleme

ile fiziksel ortamda veri ambarı oluşturulmadan önce gerekli tüm bilgileri içeren model geliştirilir. Son olarak, fiziksel modelleme ile veri tabanında ilgili veri tabanı nesnelere ve nesnelere arasındaki ilişkiler oluşturulur.

Veri modelleme aşaması veri ambarı geliştirme sürecinin önemli bir parçasıdır. Bu aşamada yapılan bir hata doğrudan veri sunum katmanını etkiler. Eksik ya da hatalı yapılan bir modelleme iş ihtiyaçlarının gerçekleştirilmesi için gerekli olan bir verinin sağlanamamasına sebep olur. Bu durum veri ambarının kullanıcı amaçlarına tam olarak hizmet edemediği için yetersiz ve başarısız olarak değerlendirilmesine sebep olur.

Geleneksel yaklaşımda veri modelleme aşamaları birbiri ile tam entegre olarak geliştirilmez. Veri modelleme aşamaları sonucunda üretilen çıktılar diğer modelleme aşamalarına doğrudan girerek kullanılmaz. Ayrıca, kavramsal modelleme aşamasından sonraki modelleme aşamaları teknik alan bilgisi gerektirdiği için iş kullanıcıları modelleme çalışmasından uzaklaşır. Bu durum, model geliştirme sırasında yapılan hataların çok ileriki aşamalarda fark edilmesine yol açar. Modelleme aşamaları arasında tam entegrasyon olmadığı için tespit edilen hataların düzeltilmesi için modelleme aşamalarının ayrı ayrı tekrar yapılması gerekir ve bu durumda geliştirme süresini uzatır.

Dinamik veri ambarı geliştirme yaklaşımı ile veri modellemenin daha hızlı ve daha entegre olması hedeflenir. Ayrıca, iş kullanıcılarının tüm veri modelleme geliştirme sürecinde yer alması sağlanarak, olası modelleme hatalarının azaltılması ve daha erken bir aşamada tespit edilerek düzeltilmesi hedeflenir.

Dinamik veri ambarı geliştirme yaklaşımında veri modelleme tek aşamada gerçekleştirilir. Kavramsal, mantıksal ve fiziksel modellemeleri birbiri ile entegre hale getirerek tek bir modelleme ortamında tüm modelleme aşamaları gerçekleştirilir. Dinamik yaklaşımda kavramsal, mantıksal ve fiziksel modelleme unsurları bulunmaya devam eder. Fakat her biri için ayrı ayrı birbirinden bağımsız modellemeler yapmak yerine tüm modelleri içeren tek bir model geliştirilir. Veri modelleme bu amaç için geliştirilmiş özel veri otomasyon uygulamaları ile gerçekleştirilir. Dinamik veri modelleme sağlayan bu uygulamalar ile iş kullanıcıları ile birlikte kavramsal bir model geliştirilir. Sonrasında aynı kavramsal model detaylandırılarak mantıksal model benzeri bir model oluşturulur. Fakat bu aşamada daha önce oluşturulmuş kavramsal model

kullanılmaya devam eder. Mantıksal model geliştirimi kavramsal modelleme sonucu oluşan nesnelere birbiri ile ilişkisi ve içerdikleri alanlar eklenerek yapılır. Tüm modelleme geliştirimi aynı önyüz kullanılarak yapılır. Kullanıcılar isterlerse her bir modelleme nesnesinin detayına bakabilir ya da detayına bakmadan genel yapıyı görebilir. Dinamik yaklaşım ile geliştirilen veri modelleme uygulamaları, modelleme önyüzlerinde gerekli tüm bilgileri girdikten sonra fiziksel modeli oluşturmaya hazır hale gelir. Fiziksel veri modelinin oluşması tamamen otomatik olarak gerçekleşir. Ayrıca, geliştirilen model teknolojik platformdan bağımsızdır. Modelleme tamamlandıktan sonra fiziksel modelin oluşturulacağı teknolojik platforma göre sadece fiziksel model özelleştirilir. Böylelikle aynı model farklı teknolojik platformlarda kullanılabilir hale gelir. Dinamik yaklaşım ile geliştirmiş veri modelleme uygulamaları, sahip olduğu metadatayı kullanarak fiziksel ortamda gerekli tabloları otomatik olarak oluşturur. Böylelikle geliştirimi tamamlanan ve onaylanan veri modeli tekrar insan kaynağına ihtiyaç duymaksızın fiziksel ortamda aynı şekilde oluşturulmuş olur. Böylelikle geleneksel yaklaşımda üç farklı aşamada yapılan ve her bir aşamada ayrı ayrı insan kaynağı gerektiren veri modelleme çok daha hızlı, daha güvenli ve tam entegre olarak geliştirilmiş olur. Ayrıca, veri modelleme uygulamaları model versiyonlarını hem fiziksel ortam hem de fiziksel ortam öncesi ortamlar için saklar. Böylelikle yapılan değişiklikleri izlemek ve yönetmek kolaylaşır.

Sonuç olarak, dinamik yaklaşım ile geliştirimde veri modelleme aşamalarının entegre edildiği ve fiziksel veri modelinin otomatik olarak oluşturulduğu bir süreç uygulanır. Tüm modelleme aşamalarının tek bir model üzerinden ve tek bir önyüz üzerinden istenen detayda gösterilmesi ile yazılım ve iş kullanıcıların tüm sürece uçtan uca dahil olmasını sağlar. Ayrıca, model versiyonlama yapılması ve fiziksel modelin tamamen otomatik oluşması sonucu geliştirimin küçük parçalar halinde tekrarlı olarak yapılmasını olanak verir. Böylelikle geliştirim süreci daha etkin bir şekilde işler.

4.2.2 Metadeta Yönetimi

Metadeta veri ambarının tüm bileşenleri ile ilgili bilgi içerir. Veri ambarının işleyişini anlayabilmek ve veri hakkında bilgiye ulaşabilmek için metadetaya ihtiyaç duyulur. Metadeta hem son kullanıcılar için hem de veri ambarını geliştiren teknik kullanıcılar için faydalı bilgiler içerir. Metadatanın olmaması ya da eksik olması iş

verimsizliğine ve kullanım hatalarına yol açar. Bu nedenle metadata geliştirimi ve etkin bir metadata yönetim stratejisinin olması oldukça önemlidir. Geleneksel veri ambarı geliştirim yaklaşımı için de bu durum geçerlidir. Fakat veri otomasyon teknolojilerin son yıllarda gelişmesi ve etkinliğini göstermesi nedeniyle geleneksel yaklaşımda son zamanlara kadar kullanılamamaktaydı. Bu durum nedeniyle bazı aşamaların manuel kodlar ve dokümantasyonlarla ilerlediği, bazı aşamalarda önceki aşamalarla entegre olmayan uygulamaların kullanıldığı geleneksel veri ambarı geliştirim yaklaşımında metadata geliştirimi ve yönetimi zor bir süreçtir. Ayrıca, metadata geliştirimi önceki bölümlerde belirtilen zorlukları nedeniyle platform ve yöntem bağımsız zorluklar da içerir. Metadata geliştiriminde genel kabul gören bir endüstri standardının olmaması da geliştirim zorluğunu gösterir. Her veri ambarı çözümü farklı bir mimari ve farklı uygulamalar ile farklı ihtiyaçlar için geliştirilir. Bu durum ihtiyaç duyulan metadatanın da farklılaşmasına neden olur.

Dinamik veri ambarı geliştirim yaklaşımı ile geliştirim sürecinin mümkün olduğunca birbiri ile entegre olan veri otomasyon uygulamaları ile gerçekleştirilmesi planlanır. Böylelikle ihtiyaç duyulan kod ve dokümantasyonun manuel olarak yapılması yerine otomatik olarak oluşturulması hedeflenir. Böylelikle daha hızlı ve daha etkin bir geliştirim mümkün olur. Veri ambarı geliştirim süreçlerinde birbiri ile entegre olan uygulamaların kullanılması ile geliştirim süreci hızlandırıldığı gibi metadata entegrasyonu da arttırılmış olur.

Veri ambarı geliştirimi genel olarak ETL olarak tanımlanan aşamalardan oluşur. Veri kaynak sistemlerden veri ambarı ortamına taşınır, sonrasında gerekli dönüştürme işlemleri yapılır ve veri yüklenir. Sonrasında veri sunum katmanında veri kullanıma açılır. Dinamik veri ambarı yaklaşımında bu sürecin geliştiriminde birbiri ile entegre olan uygulamalar tercih edilir. Böylelikle farklı uygulamalar tarafından üretilen çıktılar diğer uygulamaların ihtiyaç duyduğu girdileri sağlar. Dinamik yaklaşımda birbiri ile entegre olan uygulamalar kullanılarak metadata geliştirim ve yönetimin daha etkin bir şekilde yapılması hedeflenir. Dinamik veri ambarı yaklaşımında metadata yönetimi ve metadata erişiminin sağlandığı özelleştirilmiş uygulamalar kullanılır.

4.2.3 Veri Sunum Katmanı ve İş Zekası Uygulamaları Entegrasyonu

Veri ambarına yüklenen veriye veri sunum katmanı üzerinden erişir. Kullanıcı iş ihtiyaçlarına göre farklı özelliklere sahip veri sunum nesnelерinin geliştirilmesi gerekir. Veri ambarında bulunan veri çeşitliliği ve kullanıcı ihtiyaçlarının farklılıkları düşünüldüğünde çok fazla veri sunum nesnesinin geliştirilmesi gerektiği görülür. Operasyonel raporlamalar, analiz, veri madenciliği ve karar destek çalışmalarında ihtiyaç duyulan veri sunum nesnelерin oluşturulması gerekir. Ayrıca, oluşturulacak veri sunum nesnelерinin iş alanına göre farklılaştırılması gerekir. Geliştirilmesi gereken veri sunum nesne sayısının fazla olması hem geliştirim sürecini uzatır hem de veri sunum nesnelерinin yönetimini zorlaştırır.

Geleneksel veri ambarı geliştirim yaklaşımında veri sunum nesneleri manuel oluşturulur. Bu durum hem geliştirim süresini uzatır hem de veri sunum nesnelерinin yönetilmesini zorlaştırır.

Veri ambarı ortamlarında veriye erişim ve veri işleme için iş zekası uygulamaları kullanılır. Çoğu iş zekası uygulaması kendi metadatası oluşturulduktan sonra dinamik raporların oluşturulmasını sağlar. Geleneksel yaklaşımda metadata entegrasyonu kısıtlıdır. Bu nedenle iş zekası uygulamalarının ihtiyaç duyduğu metadatanın manuel üretilmesi gerekir. Bu durum veri sunum katmanının geliştirim süresinin uzamasına neden olur. Ayrıca, değişen ve gelişen iş ihtiyaçlarını karşılamak için veri ambarına eklenecek verilerin veri sunum katmanına yine manuel olarak eklenmesi gerekecektir. Bu durum küçük değişiklikler için hem geliştirim süresini uzatır hem de veri sunum katmanının yönetimini zorlaştırır.

Dinamik veri ambarı yaklaşımında veri ambarına yüklenen veriye metadata üzerinden erişen uygulama önyüzleri bulunur. Dinamik veri ambarı yaklaşımında veri modelleme yapıldıktan sonra fiziksel model otomatik olarak oluşturulur. Dinamik veri ambarı ortamında mantıksal veri modeli ile fiziksel veri modeli metadatası birbiri ile entegredir. Veri modeli metadataalarını kullanan bazı uygulamalar veri ambarının görsel olarak oluşturulduğu önyüzler aracılığı ile gösterilmesini sağlar. Böylelikle kod yazmaya gerek olmadan, önyüzde yer alan fonksiyonlar kullanılarak veri sunum nesnelерinin oluşturulması sağlanır. Böylelikle veri sunum nesneleri hızlı bir geliştirilir. Ayrıca, geliştirilen veri sunum nesneleri bir kaynak altında saklandığı için yönetimi de kolaylaşır.

Dinamik veri ambarı yaklaşımında metadatanın oluşturulması ve kullanılması önemlidir. Dinamik yaklaşımda fiziksel veri ambarı metadatasının iş zekası uygulamalarının ihtiyaç duyduğu formata dönüştürülmesi hedeflenir. Böylelikle geliştirilen veri modeli önce fiziksel veri ambarı ortamında oluşturulur, sonrasında ise iş zekası uygulamasının beklediği formata dönüştürülerek aynı veri modelinin iş zekası uygulaması üzerinden de erişilebilir olması sağlanır. Tüm veri modeli iş zekası uygulamasına aktarılabilceği gibi sadece seçilen bir kısmının aktarılması da uygulanabilir.

Dinamik veri ambarında metadatanın mümkün olduğunca oluşturulması ve diğer uygulamalarla entegre edilmesi hedeflenir. Veri sunum katmanında oluşturulacak nesnelere ve iş zekası uygulamaları entegrasyonunda da metadata kullanılarak geliştirim yapılır. Böylelikle veri sunum objelerinin oluşturulması ve iş zekası uygulamaları entegrasyonu mümkün olduğunca veri otomasyonu ile gerçekleştirilir.

5. ÖRNEK BİR DİNAMİK VERİ AMBARI UYGULAMA İNCELEMESİ

5.1 Kalido Hakkında Genel Bilgilendirme

Kalido uçtan uca veri ambarı çözümleri sağlayan, kendi alanında öncü bir yazılımdır. Kalido veri ambarı ve metadata otomasyonu ile hızlı ve daha kaliteli veri sağlamayı hedefler. Sağladığı veri ambarı otomasyon çözümleri ile geleneksel veri ambarı yaklaşımına göre sekiz kata kadar daha hızlı olarak sonuç alındığını belirtir [22]. Kalido 100'den fazla ülkede aralarında dünyanın en karlı şirketlerinin de olduğu kurumlarda kullanılmıştır. BP, Procter & Gamble ve Unilever şirketleri müşterileri arasındadır [23].

Kalido bünyesinde farklı ürünler barındıran, birbiri ile entegre olarak çalışan bir ürün paketidir. Mümkün olan tüm aşamalarda dinamik yaklaşımı benimseyen veri ve metadata otomasyon teknikleri kullanılarak geliştirilmiştir. Kalido bünyesinde aşağıdaki ürünler bulunur.

- **Business Information Modeler (BIM)**

BIM veri ambarı ve metadata tasarlama, geliştirim ve yönetimi sağlayan güçlü bir uygulamadır [24]. Veri modelleme ve fiziksel veri ambarı ortamının geliştirimi için gerekli tüm bilgilerin tanımlanması BIM içerisinde yapılır ve sonrasında BIM içerisinde yer alan bir fonksiyonla fiziksel veri ambarı tamamen otomatik olarak oluşturulur. BIM iş ve yazılım ekipleri arasında ortak bir dil oluşmasını sağlayarak ihtiyaçların anlaşılması, veri ambarının geliştirilmesi ve değişikliklerin yönetilmesini iyileştirir ve hızlandırır [25]. Sağladığı önyüz ve kullanımı kolay modelleme araçları ile veri modelleme, fiziksel veri ambarının oluşturulması ve yönetimi etkin şekilde yapılır.

- **Dynamic Information Warehouse (DIW)**

BIM içerisinde geliştirilen veri modeli ile fiziksel veri ambarının otomatik olarak oluşturulması sonrasında, veri ambarına veri yüklemeye kadar olan aşamalar DIW içerisinde geliştirilir. DIW içerisinde yer alan fonksiyonlar ile geleneksel yaklaşımda yer alan ETL işlemlerinin tek bir uygulama içerisinde geliştirilmesi ve otomasyonu sağlanır. Ayrıca, veri ambarının işletilmesi, çalışan işlerin ve

hataların incelenmesi ve veri ambarının bakımı gibi işlemler DIW içerisinde yapılabilir. DIW içerisinde yer alan fonksiyonlar ile veri ambarında bulunan veriler izlenebilir ve rapor geliştirimi yapılabilir.

- **Master Data Management (MDM)**

MDM uygulaması ile hem veri ambarı metadatasının hem de veri ambarına yüklenen verinin yönetilmesi ve işlenmesi sağlanır. MDM ile veri ambarında bulunan tablo ve kolon bilgilerine ulaşılır. Böylelikle veri ambarındaki verinin tanımına ulaşılan merkezi bir kaynak görevi görür. Ayrıca, veri ambarına yüklenen veri kalitesinin artırılması için kullanılacak fonksiyonlar içerir. Kaynak sistemlerden gelen verinin tutarsız, eksik ya da geçersiz olması durumunda veride oluşan bu sorunların düzeltilmesi gerekir. MDM uygulaması ile veride bulunan sorunların tespit edilmesi ve düzeltilmesine yönelik gelişmiş kurallar oluşturulur ve veri ambarına yüklenen verinin düzeltilmesi sağlanır. Böylelikle daha tutarlı ve geçerli bilgiye ulaşılmasına katkıda bulunulur.

- **Universal Information Director (UID)**

UID uygulaması DIW ile entegre olarak çalışarak iş zekası uygulamalarına veri ambarı modelinin aktarılmasını sağlar. UID uygulaması ile SAP Business Objects Designer, IBM Cognos Framework Manager ve Microsoft SQL Server Analysis Services uygulamalarına aktarım yapılabilir [26]. UID ile veri modeli metadatası iş zekası uygulamalarının beklediği formatta metadatasıya dönüştürülür ve bu işlem tamamen otomatik olarak gerçekleştirilir. Böylelikle, iş zekası uygulamaları veri ambarı nesnelere ve aralarındaki ilişkiyi tanımlayan bilgiye ulaşmış olur. Sonuç olarak, kullanıcılar iş zekası uygulamaları ile veri ambarı nesnelere ulaşarak dinamik rapor ve analiz çalışmaları yapabilirler.

Kalido'nun dinamik yaklaşımı ile geleneksel yaklaşımdan farklılaştığı özellikleri Tablo.5. ile gösterilmiştir [27].

Tablo.5. Kalido Yaklaşımının Geleneksel Yaklaşımından Farkları

Özellik	Kalido	Geleneksel Uygulamalar
Yükleme öncesi gerekli ara tabloların oluşturulması	Otomatik	Manuel
Tarihsel verinin biriktirilmesi için ihtiyaç duyulan işlemler (Slowly changing dimensions)	Otomatik	Manuel
Fiziksel şemanın yönetilmesi ile normalize verinin raporlama amaçlı denormalize edilmesi	Otomatik	Manuel
Grafiksel önyüz ile karışık veri taşıma sorgularının oluşturulması	Otomatik	Manuel
Veri yükleme, tablolarda index oluşturma, tablo ve kolon isimlendirme yönetimi	Otomatik	Manuel
Değişen kayıtların tespit edilmesi, veri doğrulama, tablo anahtar bilgilerinin yönetilmesi	Otomatik	Manuel
Veri yükleme sırasında karşılaşılan sorunların giderilmesi	Otomatik	Manuel
Arşivleme, model ve veri geri yükleme, veri denetimi ve loglama	Otomatik	Manuel
Veri tutarlılığının sağlanması, sorunların kullanıcı önyüzünden izlenmesi	Otomatik	Manuel
İş zekası uygulamaları için metadata yönetimi ve iş zekası uygulamaları ile entegrasyon	Otomatik	Manuel
Veri modeli taşıma, versiyonlama ve değişiklik yönetimi	Otomatik	Manuel

5.2 Geleneksel Yaklaşımda Karşılaşılan Sorunların Kalido ile Geliştirimi

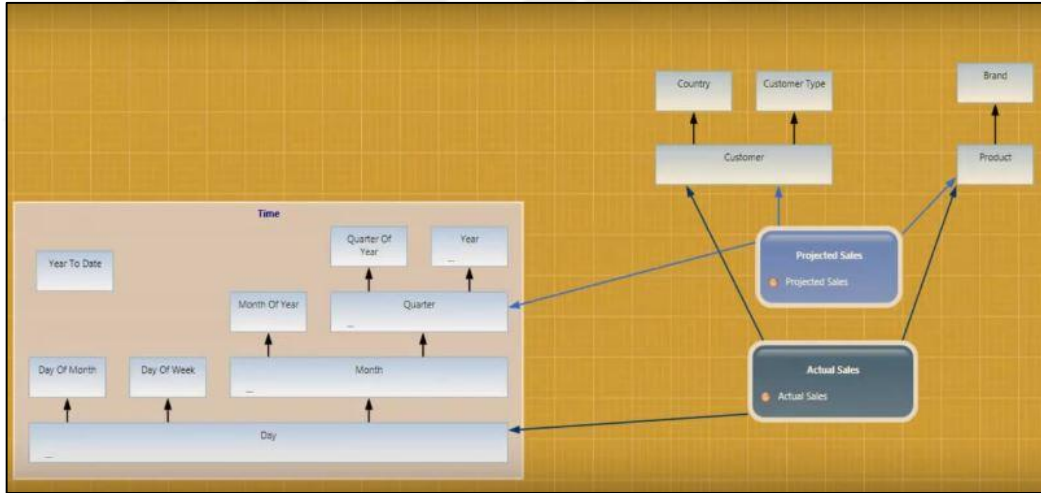
Geleneksel yaklaşım ile veri modelleme, metadata yönetimi ve veri sunum katmanının oluşturulması aşamalarında bulunan sorunların dinamik veri ambarı yaklaşımı ile geliştirilmiş veri ambarı otomasyonu sağlayan uygulamalar ile çözülebileceği önceki bölümlerde aktarıldı. Kalido ile bu aşamaların nasıl geliştirildiği uygulamaya ait ekran görüntüleri ile aşağıda gösterilmiştir.

- **Veri Modelleme**

Kalido'da veri modelleme BIM uygulaması ile yapılır. Ayrı ayrı kavramsal, mantıksal ve fiziksel modelleme aşamaları yoktur. Tek bir modelleme önyüzü kullanılarak hem kavramsal hem de mantıksal model oluşturulur. Modelleme aşamasında iş kolu ve yazılım ekipleri birlikte çalışır. Fiziksel veri ambarının oluşturulması için ayrı bir çalışma yapılması gerekmez. Uygulama içerisinde geliştirilen model ve özellikleri fiziksel veri ambarında otomatik olarak oluşturulur. Böylelikle model tamamlandıkça parça parça fiziksel veri ambarının oluşturulması sağlanır ve fiziksel veri ambarı üzerinde geliştirimler paralel olarak yürütülebilir.

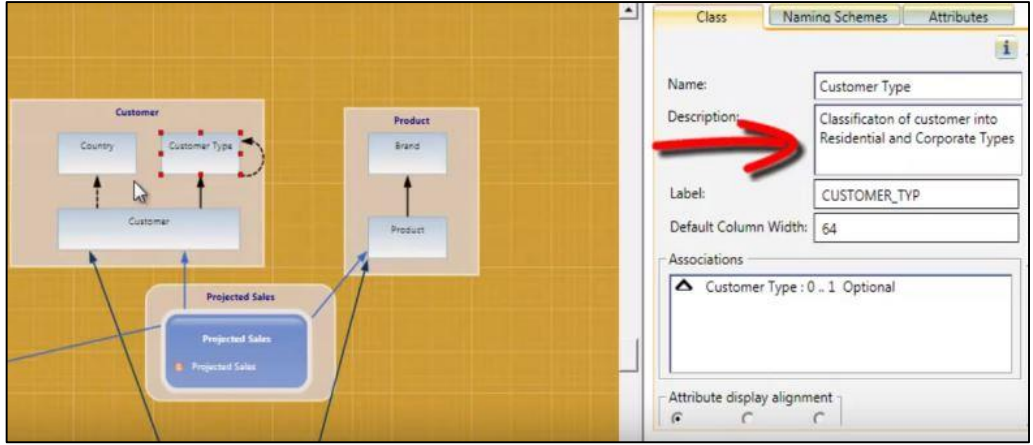
BIM ile modelleme aşamasında girilmiş tüm bilgiler metadata olarak saklanır ve veri ambarının geliştirim süresinde diğer uygulamalar tarafından kullanılır.

BIM ile veri modelleme ve fiziksel veri ambarının oluşturulması ile ilgili uygulama ekran görüntüleri aşağıda gösterilmiştir.



Şekil.14. BIM İle Geliştirilmiş Basit Bir Model

Modelde bulunan nesnelere ait detay bilgiler ekranın sağındaki panel kullanılarak girilir. Sadece teknik bilgiler değil aynı zamanda nesnenin iş fonksiyonu ile ilgili bilgiler de girilir. Bu bilgiler metadata olarak saklanır.

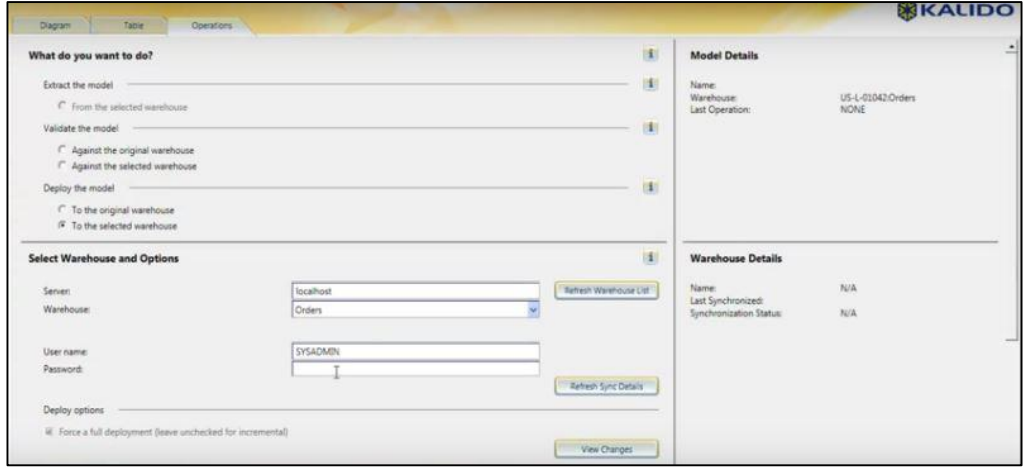


Şekil.15. BIM Modelleme Ekranı-1



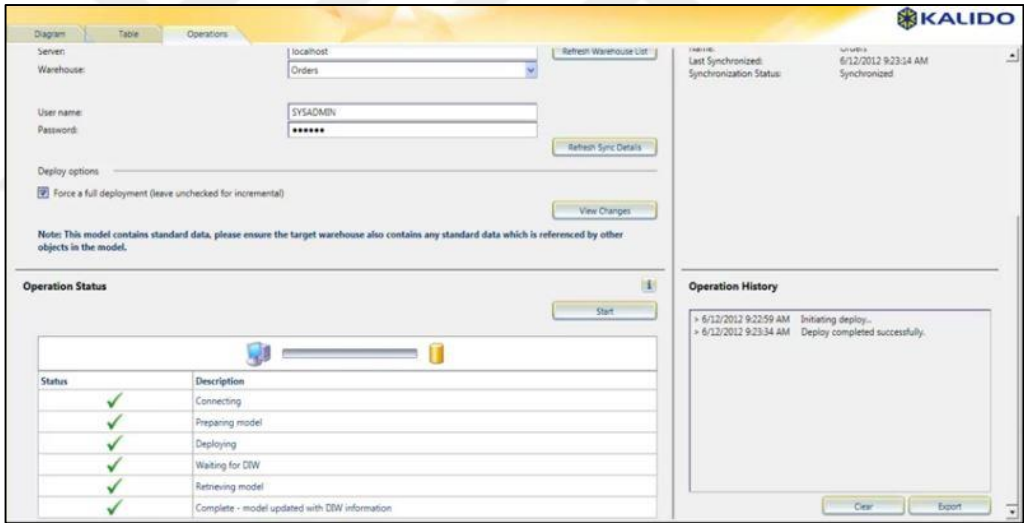
Şekil.16. BIM Modelleme Ekranı-2

BIM modeline göre fiziksel veri ambarının oluşturulması (deploy) aşağıdaki önyüz ile gerçekleştirilir. Dakikalar içerisinde fiziksel veri ambarının oluşturulması tamamlanır. Manuel yöntemler kullanılması durumunda, modelin büyüklüğüne göre, tamamlanması günler veya haftalar süren bir işlemdir.



Şekil.17. BIM Fiziksel Veri Ambarı Oluşturma (Deploy) Ekranı

Fiziksel veri ambarının başarıyla tamamlandığını gösteren ekran. Ekranın altında ve sağındaki panellerde işlemlerin durumu ve loglar gösterilir.



Şekil.18. BIM Fiziksel Veri Ambarı Oluşturma (Deploy) Sonuç Ekranı

- **Metadata Yönetimi**

BIM uygulamasında modelleme aşamasında girilen tüm bilgiler metadata olarak kayıtlanır. Bu bilgiler hem teknik kullanıcıların hem de iş kullanıcılarının veri hakkında bilgi almasını ve veriyi yönetmesi için kullanılır.

MDM uygulaması ile metadata ve veri yönetimi yapılan önyüz aşağıda gösterilmiştir.

BIM’de geliştirilen modele göre oluşturulan fiziksel veri ambarı nesnelere ve veri ambarına yüklenmiş örnek verilere MDM uygulaması ile erişilir. Veri ambarını nesnelere tanımlama ve özellikleri buradan izlenir. Bu özellikleri nedeniyle veri sözlüğü görevi görür.

The screenshot shows the Kalido MDM Product interface. At the top, there is a navigation menu with options like File, Edit, View, Favorites, Tools, and Help. Below the menu, there is a search bar and a list of suggested sites. The main content area is divided into two parts: a diagram and a table.

The diagram illustrates the product hierarchy. At the bottom, there are two boxes labeled 'Rating Agency' and 'Source Product'. Arrows point from these boxes to a central box labeled 'Product'. From the 'Product' box, arrows point to five boxes above it: 'Packaging', 'Product Class', 'Product Subgroup', 'Rebate', and 'Supplier'.

Below the diagram, there is a section titled 'Product' with a definition and a list of attributes. The definition is 'Products we sell'. The attributes listed are: Product Name, Product Code, Packaging, Rebate, Product Class, Product Subgroup, Supplier, Dfll-Source Product, Source Product Code, Dfll-Rating Agency, Avg Retail Price, Safety Rating, Suggested Retail Price, Product Weight, and Agency UPC.

At the bottom of the screenshot, there is a table with the following columns: Product Name, Packaging, Product Class, Supplier, Source Product Code, Avg Retail Price, Safety Rating, Suggested Retail Price, Product Weight, and Agency UPC. The table contains five rows of data:

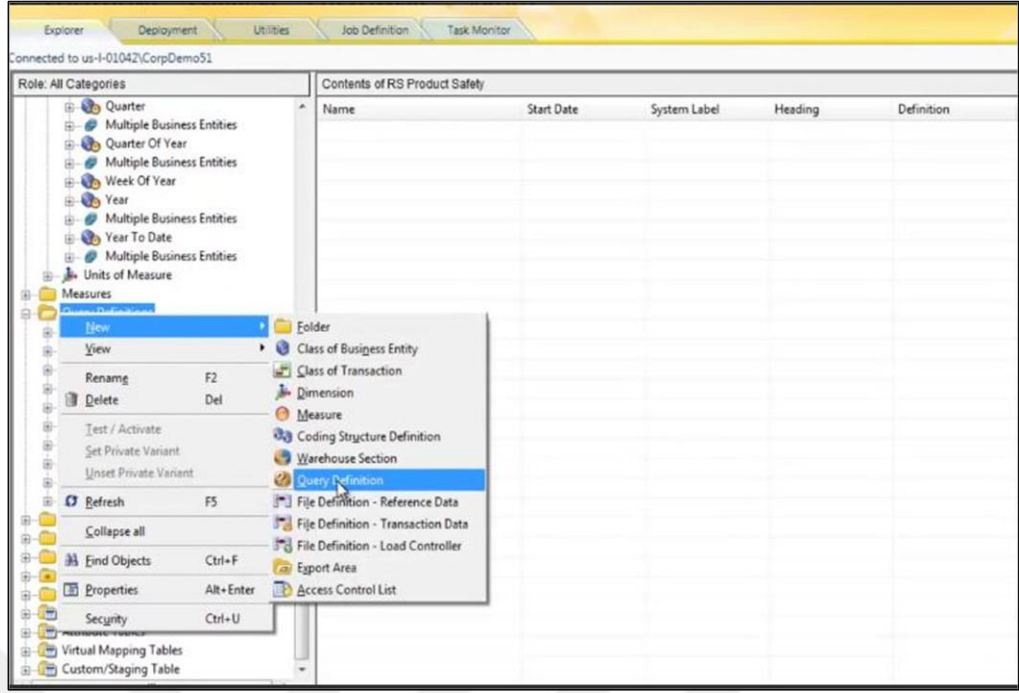
Product Name	Packaging	Product Class	Supplier	Source Product Code	Avg Retail Price	Safety Rating	Suggested Retail Price	Product Weight	Agency UPC
13 2-HEAD TV/CR COMBO*	✓	✓	✓	✓	✓	5	✓	✓	✓
13 in. Color TV 12 Volt	✓	✓	✓	✓	✓	5	✓	✓	✓
14 LCD 4:3 TV*	✓	✓	✓	✓	✓	5	✓	✓	✓
14-FLAT ANALOG STEREO TV*	✓	✓	✓	✓	✓	5	✓	✓	✓
15 SUBWOOFER*	✓	✓	✓	✓	✓	5	✓	✓	✓

Şekil.19. MDM Ekran Görüntüsü

• Veri Sunum Katmanı ve İş Zekası Uygulamaları Entegrasyonu

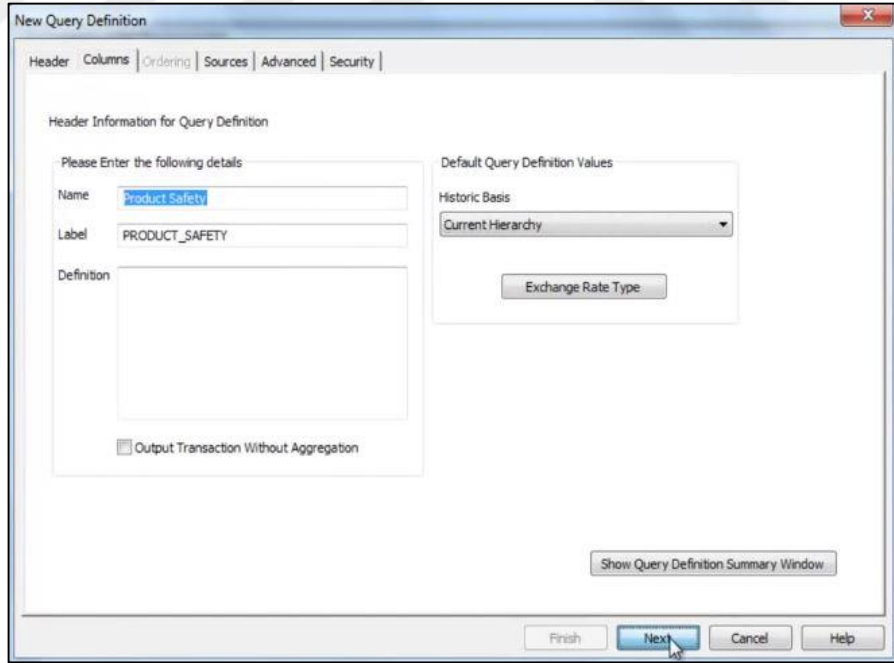
Kullanıcılar veri sunum katmanı üzerinden veriye ulaşırlar. Farklı ihtiyaçlara yönelik veri sunum katmanında birçok veri tabanı nesnesi oluşturulur. Manuel geliştirim hem zaman alır hem de yönetilmesi zordur. Ayrıca, iş zekası uygulamalarının veri ambarı ile entegre çalışması için kendi metadatasının oluşturulması gerekir. Geleneksel yaklaşımda manuel olarak geliştirilen bu aşamalar Kalido ile otomatik olarak geliştirilir. Böylelikle yönetimi de kolaylaşır.

DIW uygulamasında “Query Definition” olarak isimlendirilen fonksiyon ile veri sunum nesnelere oluşturulur.



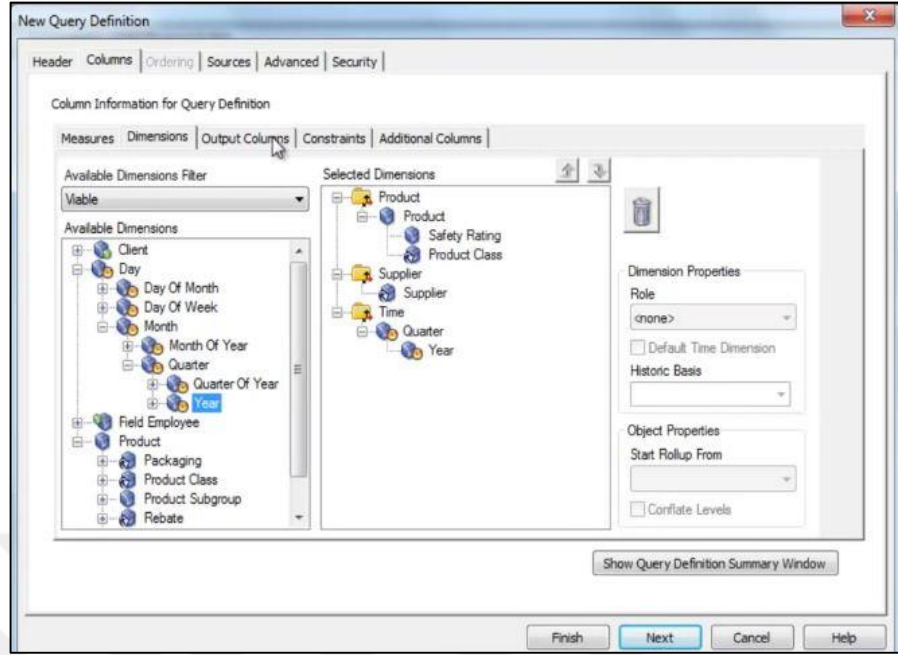
Şekil.20. DIW Ekran Görüntüsü

Query Definition önyüzünde veri ambarı metadata bilgileri kullanılarak veri sunum nesnesinin özellikleri ve kriterleri girilir.



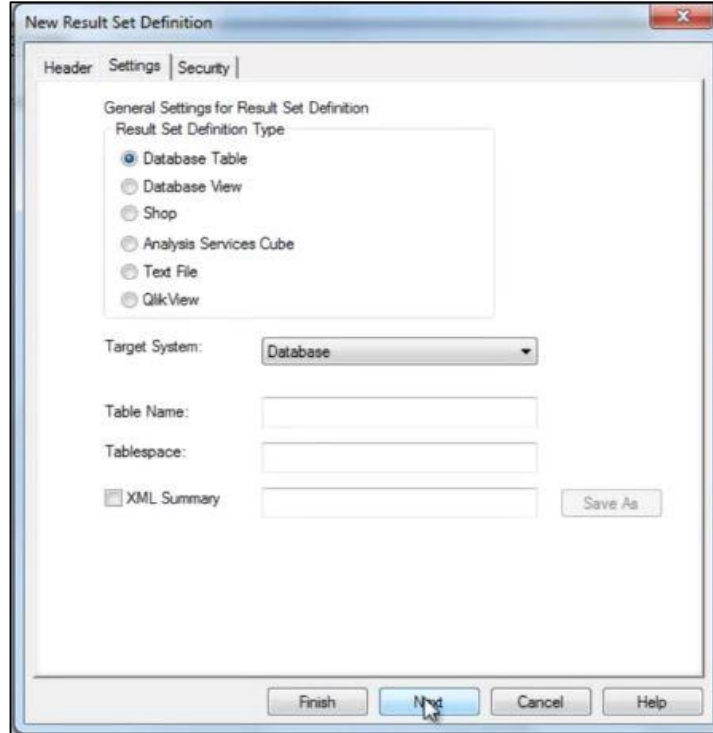
Şekil.21. Query Definition Geliştirim Ekranı-1

BIM modelinde bulunan nesnelere kullanılarak tanımlama yapılır.



Şekil.22. Query Definition Geliştirim Ekranı-2

Query Definition tanımlandıktan sonra farklı veri tabanı nesnelere ya da excel raporları oluşturmak için kullanılır. Bu şekilde benzer nesnelere tek bir kaynaktan otomatik olarak oluşturulması sağlanır.



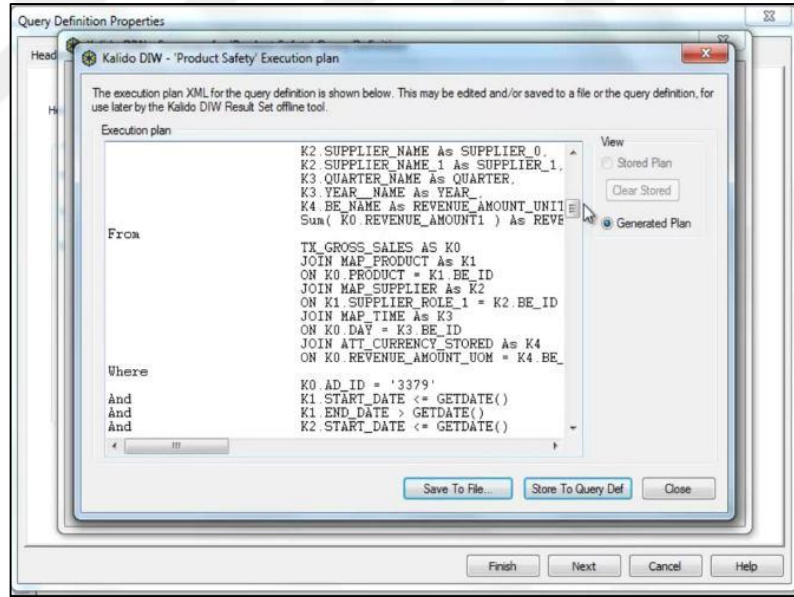
Şekil.23. Query Definition Geliştirim Ekranı-3

Excel çıktısı olarak çalıştırılmış bir query definition örneği.

Revenue Amount (Units)	(All)					
Year	(All)					
Supplier_0	(All)					
Supplier_1	(All)					
Product Class_1	(All)					
Product Class_2	(All)					
Product	(All)					
Sum of Revenue Amount		Quarter				
Product Class_0	Y	Safety Rating	2011Q1	2011Q2	2011Q3	2011Q4
- Consumables	Excellent		132,329.91	113,592.01	119,713.54	131,464.83
Consumables Total			132,329.91	113,592.01	119,713.54	131,464.83
- Durable Goods	Average		2,876.19	2,962.64	4,781.28	3,019.63
	Excellent		838,716.86	893,597.04	772,890.43	887,387.99
	Very Good		12,096.47	14,274.55	9,894.86	32,101.61
Durable Goods Total			853,689.52	910,834.23	787,566.57	922,509.23

Şekil.24. Query Definition İle Üretilen Rapor

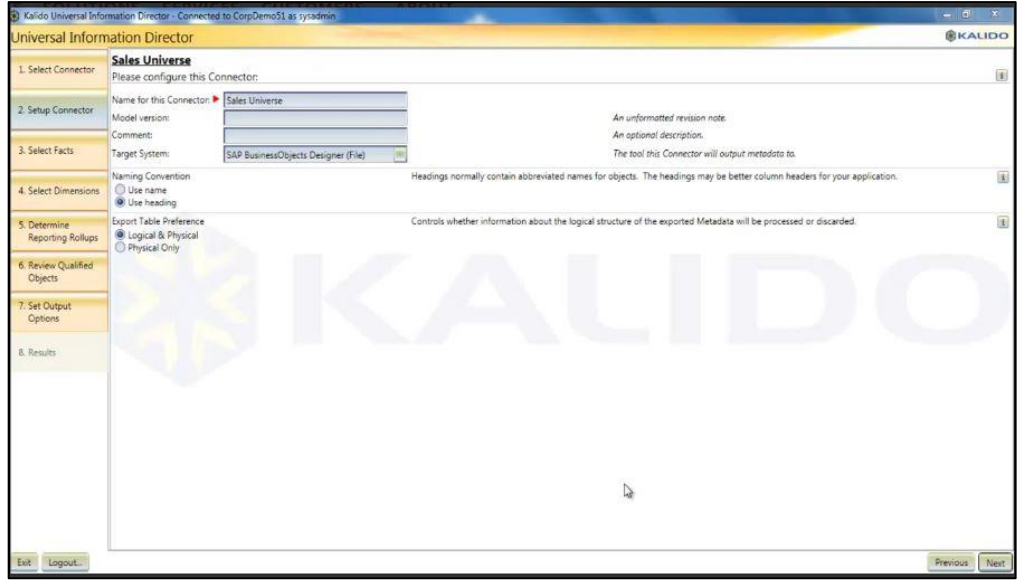
Query definition tanımına göre çalışması gereken sql kodları otomatik olarak üretilir ve ihtiyaç duyulması halinde uygulama içerisinde görüntülenebilir.



Şekil.25. Query Definition İle Üretilen SQL Sorgusu

İş zekası uygulamalarının ihtiyaç duydukları metadata bilgisi UID uygulaması ile oluşturulur. Manuel geliştirim ile günler, haftalar süren işlem bu uygulama ile dakikalar içerisinde gerçekleştirilir.

Metadata oluşturulacak hedef sistem ve diğer özellikler girilir.



Şekil.26. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-1

İş zekası uygulaması için oluşturulan metadata BIM’da üretilen nesnelere ihtiyaç duyulanların seçilmesi ile belirlenir. Böylelikle kullanıcılara sadece ihtiyaç duydukları ya da görmeye yetkili oldukları veriye erişmeleri sağlanır.

2. Setup Connector	Altı	Facts	Type	Measures	Attributes	Fact Description
	<input type="checkbox"/>	Annual Average Exchange Rate	Transaction	<input type="checkbox"/> Exchange Rate		Exchange rates
	<input type="checkbox"/>	Cash	Transaction	<input type="checkbox"/> Amount		
	<input type="checkbox"/>	Consolidated Liabilities	Transaction	<input type="checkbox"/> Amount		A consolidated feed of mi
	<input type="checkbox"/>	Daily Average Exchange Rates	Transaction	<input type="checkbox"/> Exchange Rate		Daily Average Exchange
	<input type="checkbox"/>	Gross Sales	Transaction	<input checked="" type="checkbox"/> REVENUE_AMOU		
	<input type="checkbox"/>	Loans	Transaction	<input type="checkbox"/> Amount		Commercial credit, mortg
	<input type="checkbox"/>	Margin Deductions	Transaction	<input checked="" type="checkbox"/> REVENUE_AMOU		
	<input type="checkbox"/>	Payroll Records	Transaction	<input type="checkbox"/> PAYROLL_TAXES <input type="checkbox"/> MISCELLANEOUS <input type="checkbox"/> GROSS_PAY		Record of semi-monthly p
	<input type="checkbox"/>	Portfolio	Transaction	<input type="checkbox"/> Amount		Trades of stocks and bonc
	<input type="checkbox"/>	Quarterly Average Exchange Rates	Transaction	<input type="checkbox"/> Exchange Rate		Quarterly average interba
	<input type="checkbox"/>	Sales Projections	Transaction	<input type="checkbox"/> PROJECTED_REVI		
	<input type="checkbox"/>	Summarized Payroll	Transaction	<input type="checkbox"/> Gross Pay <input type="checkbox"/> Miscellaneous De <input type="checkbox"/> Payroll Taxes		Payroll Records summariz
	<input checked="" type="checkbox"/>	Summarized Sales	Transaction	<input checked="" type="checkbox"/> Revenue Amount		
	<input checked="" type="checkbox"/>	RSV - Product Sales View	Query Definition	<input checked="" type="checkbox"/> Revenue Amount <input checked="" type="checkbox"/> Revenue Amount		

Şekil.27. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-2

	All	Dimension	CBE	Attributes	Sourced From	Prefer Attribute Tabl
2. Setup Connector	<input type="checkbox"/>					
3. Select Facts	<input checked="" type="checkbox"/>	Product	Packaging		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
4. Select Dimensions	<input checked="" type="checkbox"/>	Product	Product		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
5. Determine Reporting Rollups	<input checked="" type="checkbox"/>	Product	Product Category		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
6. Review Qualified Objects	<input checked="" type="checkbox"/>	Product	Product Class		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
7. Set Output Options	<input checked="" type="checkbox"/>	Product	Product Group		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
8. Results	<input checked="" type="checkbox"/>	Product	Product Subgroup		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
	<input checked="" type="checkbox"/>	Product	Safety Rating		Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>
	<input checked="" type="checkbox"/>	Rebate	Rebate	<input checked="" type="checkbox"/> Effective Term St. <input checked="" type="checkbox"/> Effective Term En <input checked="" type="checkbox"/> Rebate Rate	Gross Sales Margin Deductions Summarized Sales	<input type="checkbox"/>

Şekil.28. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-3

Gerekli diğer bilgilerin de girilmesi sonucu metadata oluşturulmaya hazır hale gelir.

Universal Information Director - Connected to CorpDemo51 as sysadmin

Sales Universe

Please finalize the Connector configuration:

1. Select Connector

2. Setup Connector

Name for this Connector: Sales Universe

Model version:

Comments:

3. Select Facts

Target System: SAP BusinessObjects Designer (File)

Naming Convention

Use name

Use heading

4. Select Dimensions

Export Table Preference

Logical & Physical

Physical Only

5. Determine Reporting Rollups

6. Review Qualified Objects

Please configure the selected bridge:

7. Set Output Options

8. Results

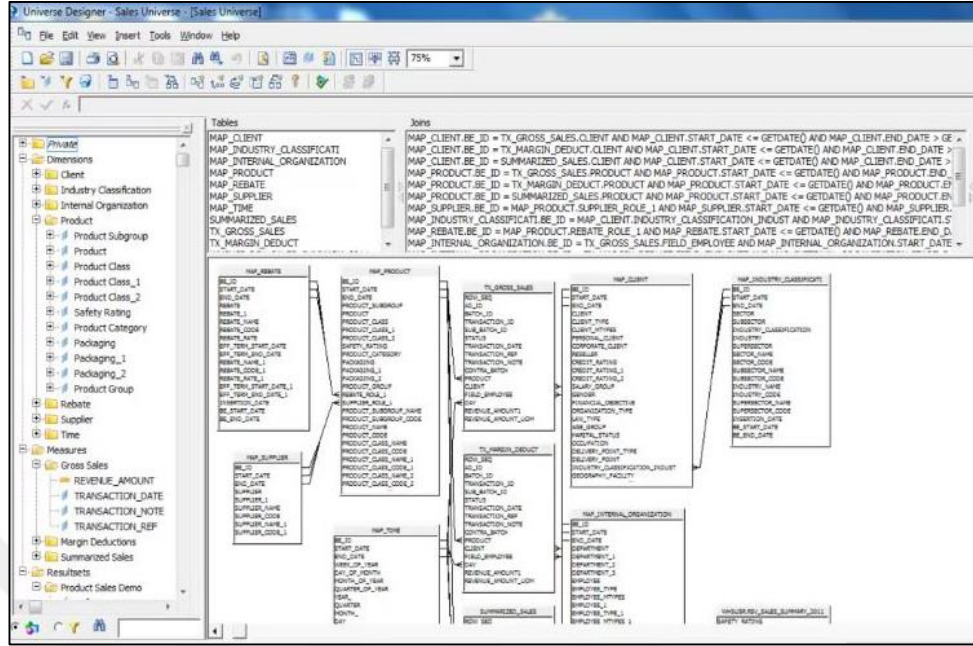
Option	Value	Description
File	C:\Users\brian.jones\AppData\...	Specify here the path to the universe file.
Data model connection to RDBMS	CorpDemo51	Specify the Connection name as it was defined in Business Objects...
System		Enter here the name of the Business Objects repository to login to.
Authentication mode	Standalone	Select the login authentication mode to be performed.
User name		A repository installation of Business Objects (BO) requires the user to...
Password		A repository installation of Business Objects (BO) requires the user to...
Data model (Table/Join) algorithm	Create all and Refresh from da...	Specify whether Tables and Joins should be exported to the universe.
Data model Outer Joins		Most modeling and ETL tools only define the metadata of foreign...
Data model layout scale	Grow	When hierarchical information is imported from another tool, it may be...

Exit Logout... Previous Run

Şekil.29. UID İle İş Zekası Uygulamasına Özel Metadata Oluşturma Ekranı-4

İş zekası uygulamasına aktarılan metadata sonrasında fiziksel veri ambarı modeli iş zekası uygulaması üzerinden erişilebilir hale gelir. SAP Business Objects (BO)

uygulanmasına metadata aktarımı sonrası veri ambarı modeli BO'da aşağıdaki gibi otomatik olarak oluşmuştur.



Şekil.30. İş Zekası Uygulamasında Oluşturulan Veri Modeli

6. SONUÇ

Günümüz dünyasında hem bireysel hem de kurumsal olarak güçlü ve sürdürülebilir bir gelişim ve büyüme için bilgiye sahip olmak gerekir. Bilgi hem bireylerin hem de kurumların entelektüel sermayesidir. Bilgi farklı biçimlerde bulunabilen, farklı kaynaklardan öğrenilebilen, üretilebilen bir varlıktır. Dijital ortamda bulunan veriler de bilginin bulunabildiği ve üretilmesinde kullanılan kaynaklardan birisidir. Teknolojik gelişme ve yaygınlaşma sonucunda teknolojik aletler ve uygulamalar hayatımızın her alanında bulunmaktadır. Bu durum dijital ortamda oluşturulan ve depolanan verinin giderek artmasına ve yönetmesi zor bir büyüklüğe ulaşmasına sebep olmaktadır. Ayrıca, dijital ortamda veriler birçok farklı formatta bulunur. Veri hacminin büyüklüğü ve farklı formatlarda bulunan verilerin birleştirilmesi gerekliliği, verinin bilgiye dönüştürülmesi sürecini zorlaştırmaktadır. Veriden anlamlı, tutarlı ve etkin bir şekilde bilgi üretebilmek için veri ambarı yaklaşımı ortaya çıkmıştır. Veri ambarı projeleri dijital verinin bilgiye dönüştürülerek katma değer yaratması için uzun yıllardan beri uygulanan geleneksel bir çözüm olmuştur. Fakat yapılan araştırmalar geleneksel veri ambarı projelerinin başarısızlık ya da sınırlı seviyede başarılı olarak sonuçlandığını göstermektedir. Veri ambarı projeleri yüksek maliyetli ve zaman gerektiren projelerdir. Projelerin yeteri kadar başarılı olmaması kurumlar için büyük bir verimsizlik ve iş kaybına sebep olur. Bu nedenle, geleneksel veri ambarı geliştirim yaklaşımında projenin başarılı sonuçlanmasını aksatan geliştirim süreçlerinin tespit edilerek iyileştirilmesi amacıyla çalışmalar yapılmaktadır. Dinamik veri ambarı geliştirim yaklaşımı da bu çalışmalar sonucunda ortaya çıkmıştır.

Dinamik veri ambarı tasarımı ile daha hızlı, daha kaliteli bilgi üretilmesi ve daha kolay yönetilebilir bir veri ambarı geliştirilmek hedeflenir. Böylelikle hızlı bir şekilde değişen ve gelişen iş ihtiyaçlarına daha dinamik bir şekilde çözüm sağlanır. Veri ambarı geliştirim sürecinde birbirinden farklı teknolojik platformlar, uygulamalar ve teknik uzmanlıklar gerekir. Geleneksel veri ambarı geliştirim yaklaşımında her aşamanın etkin şekilde geliştirilmesini sağlayan yöntemler ve uygulamalar zaman içerisinde ortaya çıkmıştır. Fakat, geliştirim aşamaları arasındaki entegrasyon sınırlıdır. Bu durum hem geliştirim süreçlerinin verimliliğini hem de veri ambarının

etkin şekilde yönetilmesini olumsuz olarak etkiler. Dinamik veri ambarı yaklaşımı ile farklı geliştirim süreçlerinin birbirine girdiler ve çıktılar sağlayarak entegre edilmesi hedeflenir. Böylelikle geliştirim süreçlerinin daha verimli olması ve veri ambarı yönetiminin daha etkin yapılabilmesi sağlanır. Dinamik veri ambarı yaklaşımında geliştirim süreci bir bütün olarak ele alınır. Mümkün olduğu sürece tüm aşamalarda birbiri ile entegre edilmiş veri otomasyonu sağlayan çözümler kullanılır. Veri otomasyonu sağlayan uygulamalar ile geliştirim sürecinin daha hızlı, daha kaliteli ve daha sorunsuz olması sağlanır. Böylelikle geleneksel yaklaşımda bulunan manuel geliştirim aşamalarında insan faktörü nedeni ile ortaya çıkan sorunların önlenmesi ve daha hızlı geliştirilmesi sağlanır.

Veri ambarı geliştiriminin önemli aşamalarından birisi de iş ihtiyaçlarının belirlenmesidir. Veri ambarı tek bir amaç ve tek bir kullanıcı grubuna yönelik geliştirilmez. Birçok amaca hizmet edebilecek şekilde veri içerir. Veri ambarı projelerinin birçoğu iş ihtiyaçlarının doğru ve yeterli anlaşılmasında sebebi ile ihtiyaç duyulan çözümleri sağlayamadığı için başarısız olarak değerlendirilmektedir. Dinamik veri ambarı yaklaşımı ile teknik ve iş kullanıcılarının proje boyunca birlikte çalışması sonucu iş ihtiyaçlarına doğru çözümler geliştirilmesi sağlanır. Bunun için iş kullanıcılarının projeye katılımı ve desteği oldukça önemlidir. Fakat iş kullanıcılar gerekli desteği verebilmesi için teknik ve iş kullanıcıları arasında ortak bir dil geliştirilmelidir. Dinamik veri ambarı yaklaşımında geleneksel veri modelleme aşamalarının iyileştirilerek veri modelinin geliştirim süresince kullanılan ortak dil olması hedeflenir.

Teknolojinin hayatımızda kapladığı alan artmaya devam ettiği sürece, bireylerin ve kurumların entelektüel sermayesini oluşturan bilginin üretilmesinde, verinin önemi artmaya devam edecektir. Kullanıcıların değişen ve gelişen iş ihtiyaçlarına dinamik veri ambarı yaklaşımı ile daha hızlı ve daha kaliteli bir şekilde çözüm üretilmesi hedeflenir. Bu hedefin gerçekleştirilmesi; geliştirim aşamalarının entegrasyonu, veri otomasyonu, teknik ve iş kullanıcıları arasında ortak bir dil geliştirilmesi ile sağlanır. Sonuç olarak, bu tez çalışması ile verinin önemi, veri ambarı geliştirim süreci, veri ambarı geliştirim sürecinde bulunan bazı aşamaların geleneksel yaklaşımda neden etkin olarak yürütülemediği ve dinamik yaklaşım ile nasıl daha etkin bir hale getirilebileceği aktarılmıştır.

KAYNAKLAR

- [1] Stamford, C., *Gartner Says More than 50 Percent of Data Warehouse Projects Will Have Limited Acceptance or Will Be Failures Through 2007*, <http://www.gartner.com/newsroom/id/492112>, 2005.
- [2] Lane, P., *Oracle9i Data Warehousing Guide Release 2 (9.2)*, https://docs.oracle.com/cd/B10500_01/server.920/a96520/concept.htm, 2002.
- [3] Inmon, W., H., *Building the Data Warehouse*, s. 31-34, 113, Wiley Computer Publishing, USA, 2002.
- [4] Guide to Data Warehousing and Business Intelligence, erişim tarihi 22.02.2016, <http://data-warehouses.net/guide/recenthistory.html>.
- [5] Columbus, L., *Roundup of Analytics, Big Data & Business Intelligence Forecasts and Market Estimates*, <http://www.forbes.com/sites/louiscolumbus/2014/06/24/roundup-of-analytics-big-data-business-intelligence-forecasts-and-market-estimates-2014/#a162cff54666>, 2014.
- [6] Kimball, R., Ross, M. (2013), *The Data Warehouse Toolkit /The Dimensional Modeling*, s. 3-4, 19, Wiley Publishing, USA.
- [7] What is Data Warehousing?, erişim tarihi 20.02.2016, <https://www.informatica.com/services-and-training/glossary-of-terms/data-warehousing-definition.html#fbid=IZzpiVYWtqT>.
- [8] How to Measure Success?, erişim tarihi 20.02.2016, <https://www.1keydata.com/datawarehousing/how-to-measure-success.html>.
- [9] Ponniah, P. (2001), *Data Warehouse Fundamentals / A Comprehensive Guide for IT Professionals*, s. 36, 64, 67, 70, 118, 193, 196, 262-263, 272-273, 279, 281, Wiley Publications, USA.
- [10] Brain, T., Benkovich, M., Dewson, R., Ferguson, S., Graves, C., Joubert, T. J., Lee, D., Scott, M., Skoglund, R., Turley, P., Youness, S., *Professional SQL server 200 data Warehouse With Analysis Services*, s. 124, Wrox Press, USA.
- [11] Glassey, K. (1998) *Seducing The End User*, Communications of The ACM, September.

- [12] George, S., *Inmon vs. Kimball: Which Approach is Suitable for Your Data Warehouse?*, erişim tarihi 02.03.2016, <http://searchbusinessintelligence.techtarget.in/tip/Inmon-vs-Kimball-Which-approach-is-suitable-for-your-data-warehouse>, 2012.
- [13] Bill Inmon vs. Ralph Kimball, 2016, Retrieved from <https://www.1keydata.com/datawarehousing/inmon-kimball.html>.
- [14] Austin, B. (2010), Kimball and Inmon DW Models, erişim tarihi 05.03.2016, <https://bennyaustrin.wordpress.com/2010/05/02/kimball-and-inmon-dw-models/>.
- [15] Breslin, M. (2004). Data Warehousing Battle of The Giants: Comparing the Basics of the Kimball and Inmon Models, *Business Intelligence Journal*, 1-20.
- [16] Ballard, C., Herreman, D., Schau, D., Bell, R., Kim, E., Valencic, A. (1998), *Data Modeling Techniques for Data Warehousing*, s. 36, IBM Press USA.
- [17] Conceptual Data Model, erişim tarihi 03.03.2016, <https://www.1keydata.com/datawarehousing/conceptual-data-model.html>.
- [18] Logical Data Model, erişim tarihi 06.03.2016, <https://www.1keydata.com/datawarehousing/logical-data-model.html>.
- [19] Mathen, M., P., Data Warehouse Testing, *Building Tomorrow's Enterprise*, erişim tarihi 20.03.2016 <https://www.infosys.com/IT-services/independent-validation-testing-services/white-papers/Documents/data-warehouse-testing.pdf>, 2010.
- [20] From Conflicting, Unintegrated Historical Data to Actionable Insight, An Introduction to Dynamic Warehousing From IBM, *Dynamic Warehousing Solutions*, erişim tarihi 20.02.2016, <ftp://ftp.software.ibm.com/software/data/bi/dynamic-whitepaper.pdf>, 2007.
- [21] Mellor, R. Traditional Data Warehouses are So Last Millennium, erişim tarihi 25.02.2016, <http://www.itproportal.com/2015/06/20/traditional-data-warehouses-so-last-millennium/>, 2015.
- [22] Why Kalido?, erişim tarihi 09.03.2016, <http://kalido.com/>.
- [23] Dramatically Accelerating the Ability To Build A Data Foundation For Analytics, erişim tarihi 05.03.2016, <http://magnitudesoftware.com/our-products/kalido/>.
- [24] Graphically Create, Share, and Maintain Business Information Models, erişim tarihi 02.03.2016, <http://kalido.com/products/business-information-modeling/>.
- [25] Kalido Business Information Modeler, erişim tarihi 02.03.2016, http://kalido.com/download/bim-datasheets/e_Kalido%20Modeler.pdf.

- [26] Kalido Universal Information Director, erişim tarihi 02.03.2016, <http://kalido.com/download/kie-datasheets/PSM-UID-91.pdf>.
- [27] Kalido is Proven to Deliver Faster than Traditional Approaches, erişim tarihi 02.03.2016, <http://kalido.com/products/kalido-information-engine/deliver-faster/>.
- [28] Physical Data Model, erişim tarihi 06.03.2016, <https://www.1keydata.com/datawarehousing/physical-data-model.html>.



ÖZGEÇMİŞ

26 Mayıs 1983 tarihinde Adana’da doğdum. İlk, orta ve liseyi Adana’da tamamladım. Lise dönemimi Adana Fen Lisesi ve Adana Gündoğdu Lisesinde tamamladıktan sonra, 2001 yılında Orta Doğu Teknik Üniversitesi Matematik bölümüne başladım ve 2006 yılında mezun oldum. 2006 yılından bu yana İstanbul’da bulunan özel sektörde faaliyet gösteren farklı kurumlarda çalıştım. Mevcut durumda uluslararası özel bir bankanın Bilgi Teknolojileri bölümünde Sistem Danışmanı olarak çalışmaya devam ediyorum.

Aday: Aytaç Bora KUTLU

TEŐEKKÜR

Gerek sađladıkları yönlendirme, gerekse her zaman olumlu yaklaşımlarından dolayı hocalarım Doç. Dr. Gökhan SİLAHTAROĐLU, Yrd. Doç. Dr. Turhan KARAGÜLER ve Yrd. Doç. Dr. Ediz ŐAYKOL'a teşekkür ederim. Ayrıca, hayatımın diđer tüm aşamalarında olduđu gibi tez yazım süresince de verdiđi destekten dolayı eşim Nilay AYTOLUN KUTLU'ya ve varlığı ile beni motive eden biricik kızım Aylin Güneş KUTLU'ya teşekkür ederim.