
GAZİ ÜNİVERSİTESİ
Gazi
1926

VERİ AMBARI OLUŞTURMADA ETL VE ELT YAKLAŞIMLARININ KULLANIMI VE KARŞILAŞTIRILMASI

Abdullah ÜNAL

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2018

Abdullah ÜNAL tarafından hazırlanan "VERİ AMBARI OLUŞTURMADA ETL VE ELT YAKLAŞIMLARININ KULLANIMI VE KARŞILAŞTIRILMASI" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Endüstri Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Öğr. Gör. Dr. Tahsin ÇETİNYOKUŞ

Endüstri Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Başkan: Prof. Dr. Hadi GÖKÇEN

Endüstri Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Üye: Prof. Dr. Ergün ERASLAN

Endüstri Mühendisliği Anabilim Dalı, Ankara Yıldırım Beyazıt Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Üye: Prof. Dr. M. Ali AKCAYOL

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Üye: Doç. Dr. Talip KELLEGÖZ

Endüstri Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Tez Savunma Tarihi: 07/06/2018

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..……

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Abdullah ÜNAL

07/06/2018

VERİ AMBARI OLUŞTURMADA ETL VE ELT YAKLAŞIMLARININ KULLANIMI VE KARŞILAŞTIRILMASI

(Yüksek Lisans Tezi)

Abdullah ÜNAL

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2018

ÖZET

Teknolojinin gelişmesiyle birlikte üretilen veri miktarı ve verinin barındırdığı bilginin değeri devamlı artmaktadır. Bu artış beraberinde verinin analiz edilebileceği veri ambarı mimarisini oluşturmayı zorunlu kılmıştır. Standard bir veri ambarı mimarisinde, veri belirli aralıklarda operasyonel sistemlerden çıkartılıp, dönüşüme tabi tutulur, temizlenir ve veri ambarına yüklenir. Ancak gün geçtikçe, güncel veri ambarına olan ihtiyaç artmış ve bu konuyla ilgili bir sürü araştırma yapılmıştır. Bu çalışma kapsamında veri ambarı konsepti hakkında bilgi verilmiş, veri ambarını oluşturan katmanlar tanımlanmış ve veri aktarımının öneminden bahsedilmiştir. Literatür araştırması kısmında, güncel veri ambarı mimarisinin hedeflendiği çalışmalar filtrelenmiş ve veri aktarımı için 2 farklı temel yöntemin kullanıldığı tespit edilmiştir. ETL (Çek Dönüştür Yükle) / ELT (Çek Yükle Dönüştür) yöntemlerinin detaylı tanımlamaları yapılmıştır. Yapılmış olan uygulamada, bu iki yöntemin belirli bir senaryoya göre veri aktarım süreleri test edilmiş ve sonuçlarının değerlendirilmesi yapılmıştır. Çıkan sonuçlara göre Çek Yükle Dönüştür yönteminin Çek Dönüştür Yükle yöntemine göre daha hızlı olduğu tespit edilmiş, güncel veri ambarı mimarisi oluşumunda önemli bir role sahip olabileceğine değinilmiştir.

Bilim Kodu	:	90607
Anahtar Kelimeler	:	Karar destek sistemleri, çek dönüştür yükle, çek yükle dönüştür, iş zekâsı, veri ambarı, veri aktarımı, ETL, ELT
Sayfa Adedi	:	45
Danışman	:	Öğr. Gör. Dr. Tahsin ÇETİNYOKUŞ

THE USAGE AND COMPARISON OF ETL AND ELT APPROACHES IN DATA WAREHOUSE MODELING

(M. Sc. Thesis)

Abdullah ÜNAL

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2018

ABSTRACT

Along with the development of technology, the amount of data produced and the value of the information it contains are continuously increasing. With this increase, the data warehouse architecture, in which the data can be analyzed, has begun to be formed. In a typical data warehouse architecture, data is periodically extracted from operational systems, transformed, cleaned, and loaded into the data warehouse. However, as time went on, the need for up-to-date data warehouses increased, and a lot of researches have been done on this subject. In this study, information about data warehouse concept is given, layers forming data warehouse are defined and the importance of data transfer is mentioned. In the literature survey, the studies targeted by the current data warehouse architecture have been filtered and it has been found that 2 different basic methods are used for data transfer. Detailed descriptions of the ETL (Extract Transform Load) / ELT (Extract Load Transform) methods have been made. In the application, the data transfer times of these two different methods are tested with respect to a specific scenario and the results are evaluated. According to the results, it has been determined that the Extract Transform Load method is faster than the Extract Load Transform method, and it is mentioned that Extract Load Transform might have a more important role within the current data warehouse architecture.

Science Code	:	90607
Key Words	:	Decision support systems, business intelligence, data transfer, data warehouse, extract-load-transfer (ELT), extract-transform-load (ETL)
Page Number	:	45
Lecturer	:	Dr. Tahsin ÇETİNYOKUŞ

TEŞEKKÜR

Yanında eğitim almaktan ve tez öğrencisi olmaktan büyük mutluluk ve gurur duyduğum, anlayışı ile bizi yalnız bırakmayan değerli hocam Dr. Tahsin Çetinyokuş'a çok teşekkür ederim. Eğitimimize büyük önem veren ve çok yönlülüğü ile desteklerini bizden esirgemeyen kıymetli hocam Doç. Dr. Talip Kellegöz'e teşekkür ederim.

Bana her zaman güven veren, hayattaki en büyük servetim olan aileme, huzur kaynağım anneme, yol göstericim babama, neşe kaynağım kardeşim Zeynep'e çok teşekkür ederim.

Değerli eşim Aysel Ünal'a hep yanımda olduğu için teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
API	Uygulama Programlama Ara Yüzü
ÇDY	Çek Dönüştür Yükle
ÇYD	Çek Yükle Dönüştür
ELT	Çek Yükle Dönüştür
ETL	Çek Dönüştür Yükle
OCI	Oracle Çağrı Ara Yüzü
SQL	Yapısal Sorgulama Dili

1. GİRİŞ

Veri toplamak ve toplanan bu veriyi anlamlandırmak insanlık tarihi kadar eskidir. En ilkel zamanlarda bile mağarada yaşayan insanlar yaşadıklarını ve tecrübelerini duvarlara çizerlerdi, bu şekilde veri birikimi olurdu. Günümüze baktığımızda ise çocuklar bile sosyal platformları, bilgisayar programlarını çok rahatlıkla kullanabilmektedirler. İşin içine teknoloji ve teknoloji ile büyüyen bir nesil girince, veri birikimi artmıştır [1]. Bizler farkında olmadan, her anımızda inanılmaz boyutta veri üretilmektedir. Her anımızı paylaştığımız sosyal medya, alışverişlerimizi gerçekleştirdiğimiz internet siteleri, kullandığımız kredi kartları, ev hayatımızı kolaylaştırmaya başlayan akıllı ev sistemleri ve bunların hepsini kontrol edebildiğimiz akıllı telefonlarımız devamlı bizlerle ilgili bilgi üretmekte ve bu bilgiler sanal ortamlarda depolanmaktadır. Çok çeşitli ortamlarda tutulan bu veriler firmalar için önemli bir bilgi kaynağı olarak kabul edilmektedir. Avrupa'nın en büyük bağımsız araştırma organizasyonu SINTEF'in raporlarına göre, günümüzde dünyada bulunan verilerin %90'ı, sadece son iki yıl içinde oluşturulmuştur [2].

Günümüzün iş dünyası gün geçtikçe bilginin öneminin farkında olmaya başlamıştır. Firmalar, geçmişe dönük verileri kullanarak mevcut durumlarını iyi analiz etmenin ve aynı zamanda geleceğe dönük yol haritası çıkarmanın yollarını keşfetmektedirler. Son zamanlarda sosyal medyada yapılan yorumların veya paylaşımların firmalar tarafında ciddiye alınarak takip edilmesi ve bu paylaşımlar üzerinden kişilere bireysel geri dönüşlerin yapılması güzel bir örnektir.

İçinde bulunduğumuz büyük veri çağında, çoğu kuruluş için veriler parçalara ayrılmış sistemler içerisinde farklı ortamlarda tutulmaktadır. Mevcut sistemler, kesintisiz veri akışını ve veri kaynaklarının karmaşıklığını yönetmekte zorlanmaktadır. Oysaki kuruluşların, iş kavrayışı için tüm veri ve içerik kaynaklarından tam olarak yararlanabilmesi gerekmektedir. Üst düzey yöneticiler, kararlarını sadece işletim verilerine veya müşterilere ait ham verilere bakarak değil, yapılandırılmış, analiz edilmiş veri veya içerikleri inceleyerek almalıdırlar. Kullanıcıların, ihtiyaçları doğrultusunda doğru bilgiye zamanında erişebilmeleri ve tüm bu verilerin yönetilebilir olması için veri ambarı modeline ihtiyaç vardır. Veri ambarı, farklı sistemlerde yer alan verilerin tek bir ortamda standart bir formatta tarihsel olarak tutulduğu ortamlardır. Standart bir veri ambarında, veriler farklı kaynak sistemlerden veri ambarına belirli zaman aralıkları ile aktarılır. Bu noktada, veri ambarının güncel olması yeni bir ihtiyaç

olarak ortaya çıkmıştır. Veri ambarının güncelliği ise veri aktarımı ile doğrudan ilişkilidir. Güncel veri ambarı ihtiyacının ortaya çıkmasıyla birlikte bu konuda yapılmış olan çalışmalar artmıştır. Bu sebeple, mevcut durumu iyi analiz edebilmek ve problemi tanımlayabilmek için literatür araştırmasına ihtiyaç duyulmuştur.

2. LİTERATÜR ARAŞTIRMASI

Araştırma yapılmak istenilen konu ile ilgili literatür araştırması yapılarak, mevcut durum analiz edilir ve böylece araştırılabilecek yeni konular keşfedilir. Literatür araştırması yapılırken, daha önce belirlenmiş olan konuyla ilgili akademik makaleler, konferanslar, bildirimler sistemli bir şekilde toparlanır.

Öncelikli olarak arama motorları ve ilgili veri tabanları taranarak yapılmış olan çalışmalar, konferanslar ve elektronik dergiler derlenmiştir. Elde edilen makaleler gözden geçirilmiş ve çalışmaya faydalı olabileceği düşünülenler daha detaylı olarak incelenmiştir. Seçilmiş olan makalelerin referans listeleri ve kaynak gösterildiği diğer çalışmalar da incelenmiştir. Yapılmış olan literatür araştırmaları sonucunda veri aktarımı için 2 farklı temel yöntemin kullanıldığı tespit edilmiş ve araştırmalar bu 2 yöntem üzerine yoğunlaştırılmıştır. Bu yöntemler Çek Dönüştür Yükle (ÇDY) ve Çek Yükle Dönüştür (ÇYD) olarak tanımlanmıştır. Bu yöntemler ilerleyen bölümlerde detaylı olarak incelenecektir ancak kısa bir tanımlama yapmak gerekirse, ÇDY, verilerin kaynak sistemlerden çıkartılması, çeşitli algoritmalara göre dönüştürülmesi ve veri ambarına yüklenmesi işlemine denir. ÇYD ise verilerin kaynak sistemlerden çıkartılması, veri ambarına yüklenmesi, çeşitli algoritmalara göre dönüştürülmesi ve sonrasında hedef tablo veya görsellere yüklenmesi işlemine denir.

2.1. Çek Dönüştür Yükle

Çalışmamızı temellendirebilmek için, Web of Science, IEEE Xplore ve Google Scholar veri tabanları başta olmak üzere ulaşılabilinen tüm kaynaklarda araştırmalar yapılmıştır. Çalışma kapsamında öncelikle, veri ambarına karşılık gelen “data warehouse” ve Çek Dönüştür Yükle (ÇDY)’ye karşılık gelen “ETL” kelimeleri anahtar sözcük olarak aranmıştır.

İlgili olduğu düşünülen çalışmalar araştırıldığında toplamda 55 tane makale ile karşılaşılmıştır. İlk çalışmanın 2003 yılında yayınlanmış olduğu gözlemlenmiştir. Makaleler başlıklarına ve özet bölümlerine göre tekrar incelendiğinde, konumuzla ilgili olabileceği düşünülen 5 tane makale irdelenmiş ve Çizelge 2.1’de makalelerle ilgili bilgiler özetlenmiştir.

Çizelge 2.1. ÇDY çalışmalarının özeti

Yazar	Yıl	Başlık	Konu	Sonuç
Simitsis, A. ve diğerleri	2005	Optimizing ETL Processes in Data Warehouses	Her bir veri aktarımı sürecinin kendisine has süreçlerinin olduğu ve bu süreçlerin mantıksal olarak optimize edilebilmesi için durum uzaylarının araştırılması konu edilmiştir. Ayrıca ÇDY süreçlerinin maliyetlerini azaltan iş akışları da geliştirilmiştir [3].	Veri aktarımı işlemlerinin, belirlenen zamanlarda yapılabilmesinin ve optimizasyonun öneminden bahsedilmiştir.
Vassiliadis, P.	2009	A Survey of Extract–Transform–Load Technology	ÇDY sürecinin konsepti ve mantıksal modellemesi hakkında bilgi verilmiştir. Her bir aşamasında karşılaşılması muhtemel problemler ve çözüm yolları öneri olarak sunulmuştur [4].	ÇDY süreçlerinin, dizayn, algoritmik ve teorik olarak geliştirilmeye açık olduğu vurgulanmıştır.
Kakish, K. ve diğerleri	2011	ETL Evolution for Real-Time Data Warehousing	Veri ambarlarının güncelliği en fazla ÇDY süreçlerine bağlıdır. Bu sebeple güncel veri ambarı ihtiyacını karşılayabilmek için ÇDY işlemine farklı bir bakış açısı getirilmeye çalışılmıştır. Belirli periyotlarla beslenen veri ambarının devamlı güncellenen bir veri ambarına dönüştürülmesi için gerekli olan mimari ve yöntemler ele alınmıştır [5].	Gerçek zamanlı iş zekâsı ve tahminleyici analizlerinin yapılabilmesi için anlık olarak veri ambarını güncelleyebilen bir ÇDY uygulamasına ihtiyaç duyulmaktadır.
Ferreira, N. ve diğerleri	2013	Near Real-Time with Traditional Data Warehouse Architectures: Factors and How-to	Geleneksel veri ambarı mimarilerini gerçek zamanlı güncel olabilmesi için uygulanması gereken yükleme stratejilerinden, yüklenecek olan verinin büyüklüğünden, indeks oluşturmaktan, veri bütünlüğü kısıtlarından ve gerçek tablo bölümlendirmesinden bahsedilmiştir [6].	Güncel veriye istenilen zamanda hızlıca erişilebilmesi için canlı veri ambarı teknolojilerinin kullanılması gerekmektedir.

Çizelge 2.1. (devam) ÇDY çalışmalarının özeti

Yazar	Yıl	Başlık	Konu	Sonuç
Sharma, S. ve diğerleri	2014	Modeling ETL Process in Data warehouse: An Exploratory Study	ÇDY yapısının detayları, veri kalitesi problemlerinin giderilmesi ve 3 farklı ÇDY modellemesinin (İfade Eşleştirme, Kavramsal Modelleme, Sınıf Diyagram (UML) Temelli) karşılaştırılması yapılmıştır [7].	Doğru analizlerin yapılabilmesi için veri kalitesinin önemli olduğu ve Kavramsal Modellemenin daha iyi olduğu sonucuna varılmıştır.

2.2. Çek Yükle Dönüştür

Benzer bir şekilde Web of Science, IEEE Xplore ve Google Scholar veri tabanları ve diğer kaynaklardan Çek Yükle Dönüştür (ÇYD)'e karşılık gelen ELT ile ilgili makaleler araştırılmıştır. ÇYD ile ilgili yapılmış olan çalışmalarda ÇDY kelimesi de anahtar kelime olarak geçmektedir. Çünkü ÇYD, ÇDY yönteminin farklı yorumlanmasıyla elde edilmiş yeni bir veri aktarma yöntemidir. Çalışmaların ortak özelliği, geliştirilmiş olan ÇYD modelinin, ÇDY yönteminden kuramsal olarak farklılıklarına ve avantajlarına değinilmiş olmasıdır.

ÇYD ile ilgili çalışmalar araştırıldığında toplamda 9 tane makale ile karşılaşılmıştır. İlk çalışmanın 2009 yılında yayınlanmış olduğu gözlemlenmiştir. Makaleler başlıklarına ve özet bölümlerine göre tekrar incelendiğinde, konumuzla ilgili olabileceği düşünülen 4 tane makale irdelenmiş ve Çizelge 2.2'de makalelerle ilgili bilgiler özetlenmiştir.

Çizelge 2.2. ÇYD çalışmalarının özeti

Yazar	Yıl	Başlık	Konu	Sonuç
Wyatt, L. ve diğerleri	2009	Principles for an ETL Benchmark	ÇDY kıyaslamaları için kıstaslara ihtiyaç duyulduğundan, kıstasların temellendirileceği senaryolardan, kaynak ve hedef veri modellerinin tanımlarından, dönüşüm çeşitlerinden ve güvenilir gereksinimlerin tanımlanmasından bahsedilmiştir. Bu kıyaslamaların hem ÇDY hem de ÇYD uygulamaları için oluşturulması gerektiği vurgulanmıştır [8].	Piyasada bulunan ve firmalar tarafından sıklıkla kullanılan veri aktarma uygulamaların, kıyaslanabilmesi için belirli kıstas ölçütlerinin oluşturulmasının öneminden bahsedilmiştir. Bu ölçütler, aynı zamanda farklı boyutlardaki veriler ve farklı senaryolarda da geçerli olmalıdır.
Freudenreich, T. ve diğerleri	2012	An On-Demand ELT Architecture for Real-Time BI	Gerçek zamanlı iş zekâsı mimarisi için 2 yöntemin uygulanmasından bahsedilmektedir. İlki, ÇDY yönteminden ziyade ÇYD yönteminin kullanılmasıyla dönüşüm işlemlerinin veri ambarı tarafında yapılmasıdır. İkincisi ise, oluşturulmuş olan veri pazarı ve görünümlerin talep edildiğinde güncellenmesidir [9].	Önerilen yöntem ile birlikte daha verimli kaynak tüketiminin gerçekleştiğine değinilmiştir. Raporlama ve analitik işlemler için veriye daha kısa sürede erişilebildiğinden bahsedilmiştir.
Waas, F. ve diğerleri	2013	On-Demand ELT Architecture for Right-Time BI: Extending the Vision	Gerçek zamanlı veri ambarı yapısından daha ziyade doğru zamanlı veri ambarı yapısının gerçekleştirilmesi gerektiği vurgulanmaktadır. Veri pazarı ve veri modellerinin ÇYD yöntemi ile ihtiyaç duyulan vakitlerde beslenmesi gerektiği ve bu aktarımların devamlı olarak takip edilmesi gerektiğinden bahsedilmiştir [10].	OLAP uygulamaları aracılığıyla talep edilen güncel veriye erişimin sağlanması ve canlı olarak akan verinin sorgu aracılığıyla kullanıcılara sunulabilmesi muhtemel sonuçlardandır.

Çizelge 2.2. (devam) ÇYD çalışmalarının özeti

Yazar	Yıl	Başlık	Konu	Sonuç
Marín-Ortega, P. M.	2014	ELTA: New Approach in Designing Business Intelligence Solutions in Era of Big Data	ÇYD yaklaşımına ek olarak ham verilerin de analiz edilerek daha hızlı çözümlerin elde edilmesi için ÇYDA (Çek-Yükl-Dönüştür-Analiz) yönteminin uygulanması gerektiğinden bahsedilmektedir. Böylece, iş zekâsı çözümlerinin dizayn edilmesinin daha kısa süreceğine, iş zekasının aynı zamanda büyük veri için de kullanılabileceğine değinilmiştir [11].	ÇYDA yöntemiyle iş zekâsı ve büyük verinin en iyi kısımları kombine edilerek iş zekasının dezavantajları ortadan kaldırılacaktır.

Araştırmalar sonucunda toplamda elde ettiğimiz 9 tane makalenin 4 tanesi sektörle ilgili dergilerde basılmıştır. Geri kalan 5 tanesi ise çeşitli konferanslarda bildiri olarak sunulmuştur.

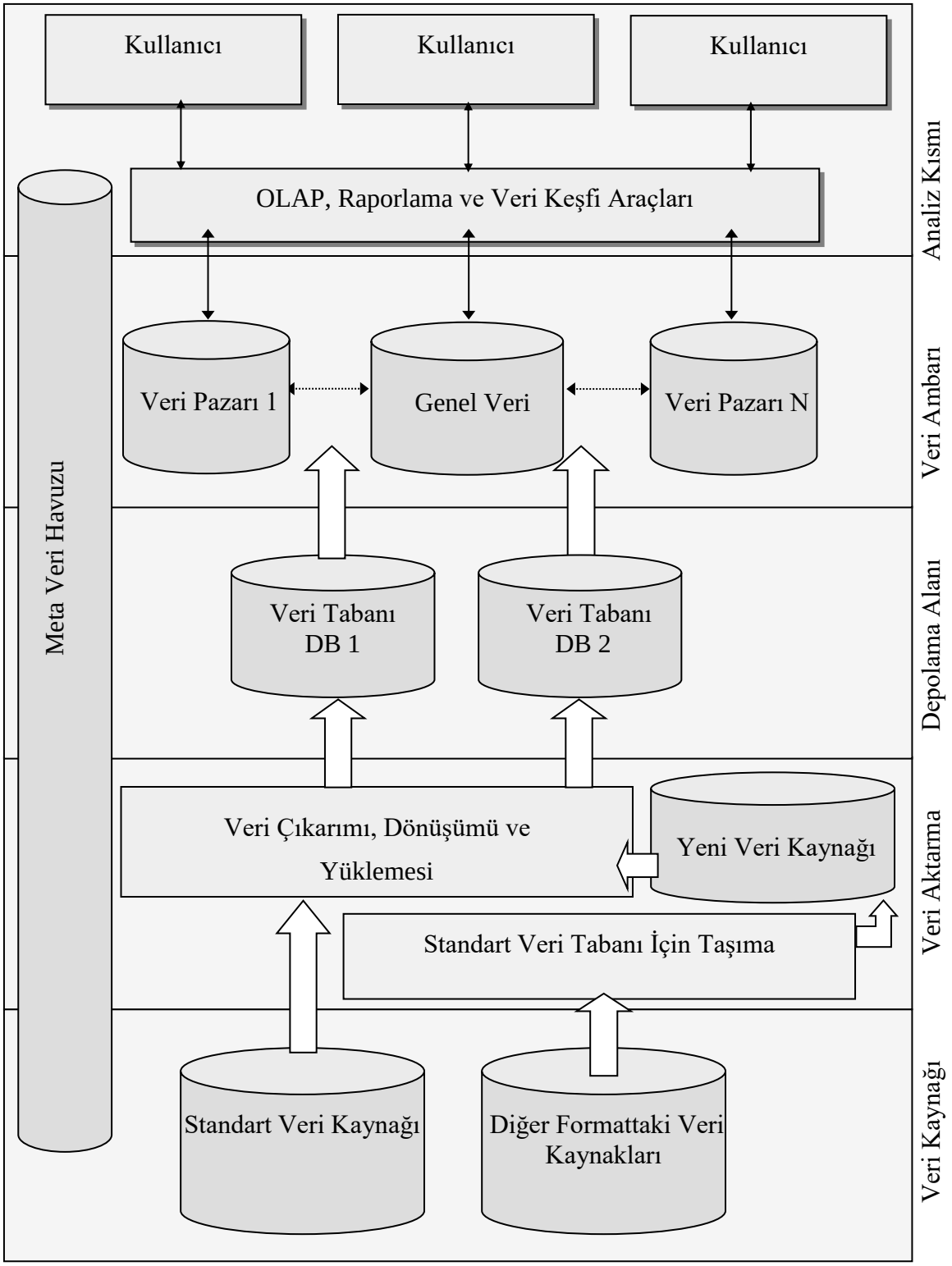
Waas ve diğerleri (2013) tarafından International Journal of Data Warehousing dergisinde yayınlanmış olan makalede, ÇDY ve ÇYD mimarilerini farklı açılardan değerlendirerek yeni bir bakış açısı sunulmaktadır ve bundan ötürü 11 makale tarafından kaynak olarak gösterilmiştir. Makale hakkında biraz daha detaylı bilgi vermek gerekirse, tipik veri ambarı mimarisinin yapısı ve işleyişi ile ilgili bilgiler verilerek giriş yapılmıştır. ÇDY yönteminin dezavantajlarından bahsedilerek gerçek zamanlı veri ambarı mimarisi için 3 tane yöntem önerilmiştir. Bunlar; ÇDY yapısının ÇYD yapısına dönüştürülmesi, veri aktarımlarının belirli periyotlardan ziyade talep edildiğinde gerçekleştirilmesi ve veri aktarımı süreçlerinin gözlemlenmesidir. Çalışmada ayrıca, gerçek zamanlı veri ambarı yapısından ziyade doğru zamanlı veri ambarı mimarisinin önemi vurgulanmaktadır. Çünkü, veri ambarına aktarılan her verinin güncel olmasına gerek olmayabilir. ÇDY ve ÇYD yapılarının karşılaştırılabilmesi için kıyaslama kriterlerinin netleştirilmesi gerektiğinden bahsedilmektedir. Son olarak, önerdikleri mimariyle ilgili 8 tane açık problem irdelenmiş ve bunlarla ilgili farklı çalışmaların ileri zamanlarda yapılabileceği belirtilmiştir [10].

ÇDY ve ÇYD ile ilgili yapılan literatür araştırması sonucunda, güncel veri ambarı mimarisinin oluşturulmasının ortak hedef olduğu gözlemlenmiştir. Bu hedefe ulaşmak için kimi zaman ÇDY yöntemi kimi zaman da ÇYD yöntemi temel alınarak farklı yaklaşımlar geliştirilmiştir. Bu çalışma kapsamında yapılması planlanan, ÇDY ve ÇYD yöntemlerinin verimlilik ve performans açısından kıyaslanmasıdır. ÇDY ve ÇYD yöntemleri ile ilgili detaylı bilgi vermeden önce veri ambarı mimarisinin iyi bir şekilde tanımlanmasına ihtiyaç vardır.

3. VERİ AMBARI

Bilgi sistemleri, teknolojinin gelişmesiyle birlikte 1960'lı yıllarda ortaya çıkmış ve sonrasında uygulanmaya, geliştirilmeye başlanmıştır. Veri ambarının hikayesi ise bilgi sistemleri ve karar destek sistemlerinin gelişimiyle birlikte başlamıştır. Veri ambarı terimini ilk olarak 1991 yılında William H. Inmon tarafından kullanılmıştır. O, veri ambarını yönetimsel karar vermeye yardımcı olacak verilerin konu odaklı, birleştirilmiş, sabit ve değişken zamanlı olarak toplanması olarak tanımlamıştır [12]. Veri ambarı, belirli bir konu üzerinde analiz yapmaya olanak sağlar. Mesela, bir raporda ilgili şirketin sadece müşterilerine ait bilgiler yayınlanırken başka bir raporda da hem müşteri hem de satış bilgileri ile ilgili bilgiler analiz edilebilmektedir. Veri ambarı, farklı kaynaklardan gelen bilgileri birleştirmektedir. Böylece farklı kaynaklarda farklı tanımlanmış olan ürün, müşteri gibi bilgiler ortak bir değerle tanımlanmış olacaktır. Veri ambarına aktarılmış olan veriler değiştirilmemektedir böylece geçmişe dönük bilgilere ulaşılabilmektedir. Veri ambarı, verilerin güncel ve geçmiş değerlerinin saklandığı veri tabanlarıdır. Canlı sistemlerde sadece güncel veriler tutulurken veri ambarında geçmişe dönük bilgiler de depolanmaktadır. Örnek olarak, bir şirketin güncel ve geçmişe dönük müşteri rakamlarına veri ambarından erişilebilir.

Hizmet sektöründen üretim sektörüne kadar birçok firma tarafından, iş zekâsı uygulamalarını oluşturmak ve geliştirmek için kullanılan veri ambarları, karar destek sistemleri olarak tasarlanmıştır. Ralph Kimball tarafından ise, analiz ve sorgu amaçlı canlı sistemlerdeki verilerin belirli bir biçimde kopyası olarak tanımlanmıştır [13]. Yapılacak analizlerin, analiz edilecek istatistiki bilgilerin veya sunulacak raporların kolay ve hızlı hazır olabilmesi için, veriler, okunmaya hazır ve hesaplanmış olarak tutulmaktadır. Bu süreçte aynı zamanda verilerin filtrelenmesi, temizlenmesi, değiştirilmesi, birleştirilmesi gibi işlemler de uygulanmaktadır.



Şekil 3.1. Veri ambarı mimarisi [14]

Veri ambarı yapısı, geliştirildiği sisteme ve amaca yönelik farklılık gösterebilmektedir. Bazı sistemlerde bir tane kaynak sistem varken başka ortamlarda da onlarca kaynak sistem olabilir. Bazıları sadece tek bir veri pazarı ile geliştirilmişken, bazıları da yüzlerce tablodan

oluşmuş veri modeli ile hizmet vermektedir. Bu sebeple veri ambarı mimarisini Şekil 3.1'de de görüldüğü üzere genel olarak 6 katmanda inceleyebiliriz.

3.1. Veri Kaynağı Katmanı

Veri kaynağı, bilgisayar uygulamalarında verinin saklandığı, güncellendiği ortamdır. Veriler disklerde veya uzak sunucularda tutulmaktadır. Veri kaynağı denilince ilk akla gelen veri tabanıdır ancak dosyalar, veri sayfaları, XML dosyaları da veri kaynağı olarak kullanılmaktadır. Veri kaynakları, kullanım yeri ve amacına göre değişiklik gösterebilir. Bilgisayar uygulamaları bir veya daha fazla veri kaynağına sahip olabilir.

Herhangi bir veri ambarı için veri kaynağı olmazsa olmazdır. Veri ambarı farklı veri kaynakları tarafından beslenmektedir. Düz metin, Excel dosyası veya ilişkisel veri tabanları veri kaynakları olarak tanımlanabilir. Bu veri kaynakları içerisinde yer alan verilerde ise, operasyonda kullanılan satış bilgileri, sisteme giriş yapan kullanıcı bilgileri, market araştırma bilgileri veya anket sonuçları yer alabilir [14].

3.2. Veri Aktarma Katmanı

Bu katmanda, veriler kaynak sistemlerden çıkarılıp depo alanında saklanmaktadır. Depo alanında yer alan veriler daha sonra dönüştürülerek veri pazarı veya tablo olarak veri ambarına yüklenir. Dönüştürme işlemi sırasında uygulanan mantık ile birlikte veriye anlam kazandırılır ve veri temizlenir. Veri ambarı geliştirmelerinde zamanın büyük çoğunluğu bu katmanda harcanmaktadır. Bazı araştırmalara göre çıkarma, dönüştürme ve temizleme sırasında harcanan iş gücü ve zaman, veri ambarı geliştirmesinin %80'nini teşkil edebilmektedir [15]. Veri aktarma katmanı, farklı veri kaynaklarının birleştirilmesi, yeni kaynakların eklenmesi ve veri aktarımının devam ettirilmesini kolaylaştıran bir katmandır [14]. Veri aktarma ile ilgili ilerleyen kısımlarda daha detaylı bir bilgilendirme yapılacaktır.

3.3. Depolama Alanı Katmanı

Kaynaktan çıkarılan veri, veri ambarına yüklenmeden önce bu katmana aktarılır. Bu katmanda, veri temizleme, değiştirme ve birleştirme işlemlerinin ilk aşamasına tabi tutulur [14]. Depolama alanı katmanı, veri ambarı projelerini kolaylaştıran ve sürekliliğini mümkün

kılan temel konseptlerden birisidir. Veri ambarını oluşturacak temel tablolar bu kısımda depolanır ve sistemin geliştirilmesinde ihtiyaç duyulmayan bilgilere yer verilmez.

3.4. Veri Ambarı Katmanı

Veri pazarlarının ve tabloların birbirleriyle belirli bir mantık içerisinden ilişkilendirildiği ve veri modelinin geliştirildiği katmandır. Boyutlu modelin kullanılmasının 2 temel avantajı vardır. Bunlardan ilki, ilişkilendirilmiş veri tabanı ortamında çok boyutlu analiz yapılabilmesine olanak sağlamasıdır. İkinci olarak, özgün ve standartlaştırılmamış boyutlu model, sorgulama sürecini kolaylaştıran ve performansı arttıran basit bir şema yapısına sahiptir [16]. Veri ambarı katmanının iyi anlaşılabilmesi için birkaç tanımlamanın yapılmasına ihtiyaç vardır.

Olgu Tablosu: Bilginin farklı boyutlarda saklandığı tablodur. Verinin tarihsel değişimi de bu tablolardan çıkartılır. Mesela, bir ürünün hangi lokasyonda, ne zaman, kim tarafından, hangi müşteriye, kaç tane satıldığı raporlanmak isteniyorsa, olgu tablosundan bu bilgilerin saklandığı kolonların olması gerekmektedir. Bu kolonlarda veriler genellikle ID şeklinde saklanır ve yabancı anahtar olarak adlandırılır.

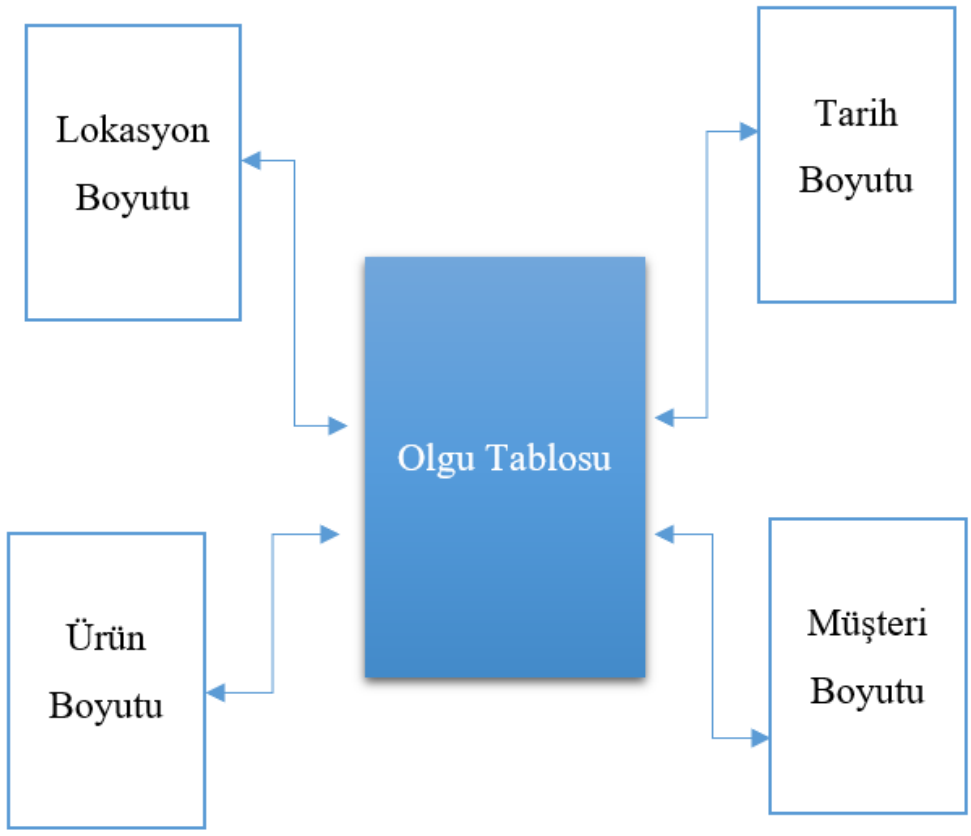
Boyut Tablosu: Olgu tablosunda yer alan bilgilerin isimlerinin, tanımlamalarının detaylı olarak tutulduğu tablolardır. Olgu tablosunda yer alan yabancı anahtarlar üzerinden bu tablolarda yer alan benzersiz ID'ler ile yani birincil anahtarlarla ilişkiler tanımlanır ve ilgili bilgiye bu şekilde ulaşılır. Mesela, olgu tablosunda şehir bilgisi "06", "34" gibi yabancı anahtarlarla tutulur ve tablo genelinde birçok kez tekrar edebilir. Ancak, boyut tablosunda birincil anahtarlar tekrar etmeksizin tutulur ve her bir birincil anahtara denk gelen şehir adı, coğrafi konumu gibi detay bilgiler boyut tablolarında yer alır. Böylece veri tabanında yer tasarruf edilmiş olur.

Ölçü: Olgu tablosunda tutulan sayısal bilgilere ölçü denilmektedir. Mesela bir ürünün satıldığı miktar ölçü olarak tanımlanabilir. İhtiyaç duyulan matematiksel hesaplamalar (ortalama, toplama, sayma, en büyük/düşük değer vb.) ölçü değerleri üzerinden hesaplanır. Ölçüler kimi zaman boyut tablosunda da tutulabilir.

Veri ambarı modellemesinde en çok kullanılan 2 farklı yaklaşım vardır.

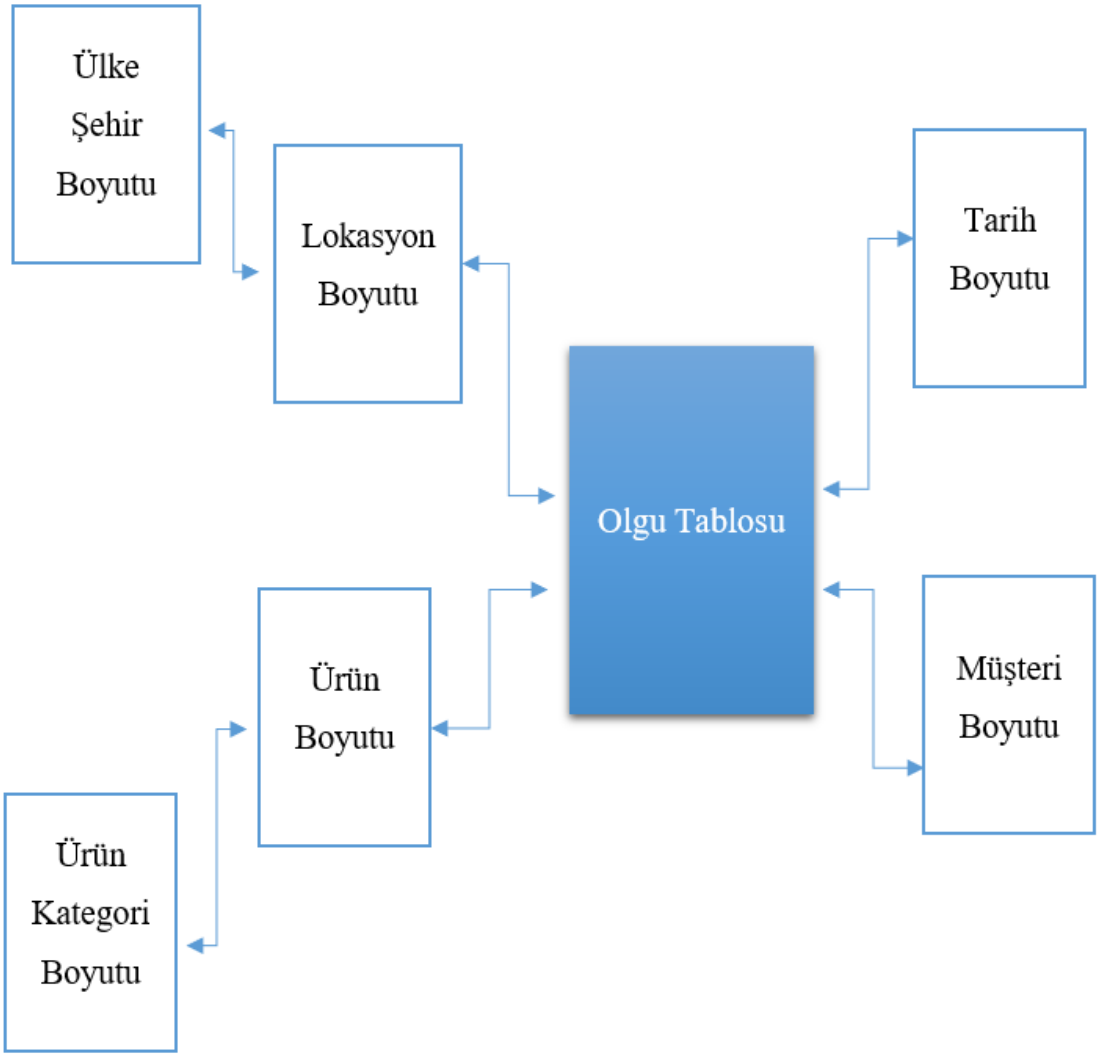
3.4.1. Boyutsal yaklaşım (Kimball metodu)

Yıldız ve Kar Tanesi şemaları veri ambarı modellemesinden en çok kullanılan boyutsal yaklaşımlardır. Yıldız veya Kar Tanesi şemalarından hangisinin tercih edileceği iş ihtiyacına göre değişmektedir.



Şekil 3.2. Yıldız şema modeli

Yıldız şema dizaynında, Şekil 3.2'de de görüldüğü üzere merkezde 1 tane olgu tablosu ve çevresinde de onunla ilişkili boyut tabloları yer alır. Olgu tablosunda yer alan her bir boyut, tek bir boyut tablosuyla açıklanabilir. Boyut tablolarında yer alan birincil anahtarlar olgu tablosundaki ilgili yabancı anahtara bağlıdır. Yıldız şeması, basit veya karmaşık olabilir. Basit star şemada sadece bir tane olgu tablosu varken; karmaşık star şemada birden fazla olgu tablosu vardır.



Şekil 3.3. Kar Tanesi Şema Modeli

Kar Tanesi şeması modeli, Şekil 3.3'teki gibi, yıldız şema modelinin biraz daha genişlemiş halidir. Bu modelde, olgu tablosunda yer alan her bir boyut, birden fazla boyut tablosuyla açıklanabilmektedir. Mesela, satılmış olan ürünün hangi ürün kategorisine ait olduğu veya satılmış olan lokasyonun hangi şehir veya ülkede bulunduğunu bulmak için boyut tablolarıyla ilişkilendirilmiş bir sonraki boyut tablosuna gitmek gerekmektedir. Bu modelin en büyük avantajı, daha fazla yer tasarrufunun sağlanması ve böylece veri tabanındaki sorgu performansının artmasıdır. Dezavantajlarından birisi de tablo sayısındaki fazlalık sürdürülebilirliği ve bakımı zorlaştırmaktadır [17].

3.4.2. Normalize yaklaşım (Inmon metodu)

Veriler daha çok veri tabanı normalizasyon kurallarına göre depolanmaktadır. Belirlenmiş olan temel konu başlıklarına göre tablolar oluşturulurlar. Mesela, siparişlerle ilgili bir tablo oluşturulduğunda, bir siparişe ait bütün detaylar bu tabloda yer almaktadır. Bunun gibi, müşteri, ürün, tedarikçi, mağaza gibi farklı konu başlıkları için farklı tablolar oluşturulur. Bu yöntemde, veri tabanına yeni bir bilgi eklemek oldukça basittir. Ancak zamanla bu model karmaşık hale gelmeye başlar ve tablo ilişkilerini sağlamak, sorgu çekmek zor hale gelebilir [18].

3.5. Analiz Katmanı

Bilgilerin kullanıcılara sunulduğu katmandır. Kullanıcılar analitik araçlarla verilere erişerek analiz yaparlar. Analitik araçlar verinin bilgiye dönüşmesine olanak sağlar. Kullanıcıların kavrama yetkinliği ile birlikte, elde bilgilerden yeni kurallar, öngörüler ve algoritmalar elde edilir.

Çevrimiçi analitik işleme (OLAP), tablo ve grafik gibi görsel içerikli raporlama uygulamaları, veri madenciliği uygulamaları en çok kullanılan analitik araç türlerindendir. Bunların dışında özetlenmiş tablolar ve kullanıcılara mail yoluyla iletilen bilgilendirmeler yine bu katmanda hazırlanmaktadır. Her bir işlem için ayrı bir sistem ve geliştirme gerekmektedir.

3.5.1. Çevrimiçi analitik işleme (OLAP)

Çevrimiçi analitik işleme, verinin her bir boyut bazında dilimlenerek küp şekline getirilmesi ve sonrasında kullanıcının istediği boyutta veriye ulaşması özgürlüğüne denir. Böylece kullanıcılar, veriye çok hızlı bir şekilde farklı bakış açılarıyla yaklaşabilirler. Mesela, bütün ürünlerin Ankara ilinde 2017 yılında gerçekleşmiş olan satış rakamlarına ulaşılabiliyorken aynı zamanda belirli bir ürünün yıllar bazında bölgesel karlılık oranlarına da ulaşılabilir. Çevrimiçi analitik işlemede veriye hızlı bir şekilde ulaşılabilmesinin sebebi hesaplamaların daha önceden yapılmış olmasıdır. Verinin en küçük biriminden bütününe kadar her aşamada hesaplamalar yapılmıştır. Bu hesaplamalar sadece toplama değil aynı zamanda saydırma, ortalama alma, yüzdesel dağılımı bulma ve sıralama olabilmektedir. Verinin çok boyutlu

incelenebilmesi, her seviyede yapılan sorgulamaların eşit sürede sonuçlanması, esnek rapor oluşturabilme ortamının olması, gruplama yapılırken herhangi bir sınırlamanın olmaması ve birden fazla kullanıcının eş zamanlı bağlanabilmesi, çevrimiçi analitik işlemeyi cazip bir analitik araç yapmaktadır.

3.5.2. Raporlama sistemleri

Raporlama araçları ile birlikte kullanıcılar istedikleri bilgiye hızlı bir şekilde erişebilmektedir. Dashboard, puan çizelgesi, otomatik raporlama ve veri görselleştirme gibi uygulamalarla daha hızlı ve hedef odaklı kararlar alınabilmektedir. Kullanım ihtiyacına göre veya kullanıcıların yetkinliğine göre farklı raporlama sistemleri tercih edilebilmektedir.

Plansız (adhoc) raporlar, bir defaya mahsus olmak üzere ihtiyaç duyulan bilgiye ulaşmak için kullanılır. Plansız raporların hazırlanabilmesi için kullanıcıların SQL dilini iyi bilmesi ve sistem verilerine hâkim olması gerekmektedir. Anlık raporlar daha çok karmaşık hesaplama ve filtreleme gerektiren talepler için üretilir.

Otomatik raporlama araçları, kullanıcıların kendi raporlarını istedikleri zamanda istedikleri kriterlerde üretebildikleri raporlama platformlarıdır. Bu platformlar anlık ihtiyaçlara cevap verebilmeli, kolay erişilebilmeli ve ara yüzleri kullanıcı dostu olmalıdır. Bu uygulamaların arka tarafında sistem yöneticileri tarafından oluşturulmuş olan bir veri modeli bulunmaktadır. Bu veri modelini küçük bir veri ambarı modeli gibi de düşünebiliriz. Raporlamada ihtiyaç duyulan tablolar bu modele eklenir ve modelde yer alan diğer tablolarla ilişkisi tanımlanır. Daha sonra, kullanıcıların ihtiyacına göre ön yüz objeleri hazırlanır ve kullanıcılar bu objeleri raporlama alanına sürükleyip bırakarak ihtiyaç duydukları verileri görebilirler. SQL diline hâkim olmayan bir kullanıcı da bu şekilde rahatlıkla rapor oluşturabilir.

Gösterge panoları ve puan kartları diğer raporlama araçlarına göre görselliğin ön planda olduğu uygulamalardır. Bu uygulamalar veriyi görsele dönüştürmede çok başarılıdırlar. Ayrıca sahip oldukları zengin sorgulama diliyle birçok matematiksel, finansal, istatistiksel, mantıksal hesaplamaları yapmaya olanak sağlarlar. Bu uygulamalar, çeşitli veri tabanlarına bağlanarak bilgiyi kendi belleklerine alabildikleri gibi kendi belleklerine almadan da direk veri tabanı üzerinden de çalışabilmektedirler. En önemli özelliklerinden birisi de analizlerin

hızlı bir şekilde yapılabiliyor olmasıdır. Bu uygulamalarda, veri kaynaklarının özellikleri, tablolar arasındaki ilişkiler, verilerin ne zaman nasıl güncelleneceği gibi bilgiler bir kez tanımlanmaktadır. Bu tanımlamalardan sonra uygulama tarafından çekilen veri, kullanıcının hazırlamış olduğu tablolar, grafikler otomatik olarak gerçekleşmektedir. Bu güncellemeler de kullanıcıların ihtiyaçları doğrultusunda günün belirli saatlerinde veya yılın belirli gün ve aylarında olacak şekilde kurgulanabilmektedir. Ayrıca bu uygulamalara mobil cihazlar veya tabletlerden rahatlıkla erişilebilmektedir.

Puan kartları stratejik raporlar iken gösterge panoları daha çok taktiksel veya operasyonel raporlardır. Bu sebeple puan kartları üst seviye yönetim kadrosuna hitap ederken gösterge panoları orta seviye yöneticiler veya performans takip eden kullanıcılara hitap eder. Puan kartlarında genelde hedeflenen değerler ve mevcut değerlerin hedeflenen değerlerin neresinde olduğu raporlanır. Bu açıdan baktığımızda gösterge panoları performans izleme; puan kartları ise performans yönetme araçları olarak düşünülebilir. Özetle, teknolojik altyapı anlamında hiçbir farkı olmayan bu uygulamalar, içerdikleri verinin anlamı ve hedef kullanıcı kitlesine göre gösterge panosu veya puan kartı olarak kullanılmaktadır.

Veri madenciliği uygulamaları, raporlama araçlarına göre daha komplekstir. Yapılacak olan çalışmanın hedefine, kullanılacak olan verinin karakteristiğine göre seçilecek ve uygulanacak olan algoritma değişiklik göstermektedir. Doğru bir analizin yapılabilmesi için ciddi bir bilgi birikimine ihtiyaç vardır. Veri madenciliği sonucunda veriler içerisindeki gizli ilişkiler açığa çıkartılır ve karar alma mercilerince değerlendirilmeye tabi tutulur.

Özetlenmiş tablolar ise sorguların çalıştırılması sonucunda elde edilir. Özel ihtiyaçlara yönelik hazırlanmış olan sorgular, çeşitli algoritmalar, fonksiyonlar ve tablolar arası ilişkiler içermektedir. Özetlenmiş tabloların en büyük avantajı, ihtiyaç duyulan bilgiye erişim süresinin diğer yöntemlere kıyasla daha kısa olmasıdır [14].

3.6. Meta Veri Katmanı

Veri ambarı mimarisi içerisinden yer alan verilerin bilgilerinin tutulduğu katmandır. Daha açık bir ifadeyle, bir bilginin hangi kaynaktan beslendiğinin, hangi dönüşüm işlemlerine tabi tutulduğunun, hangi raporlarda görselleştirildiğinin bilgisidir. Meta veri katmanı, veri ambarının verimliliği ve etkinliği için önemlidir. Çünkü iyi bir meta veri bilgisine sahip olan

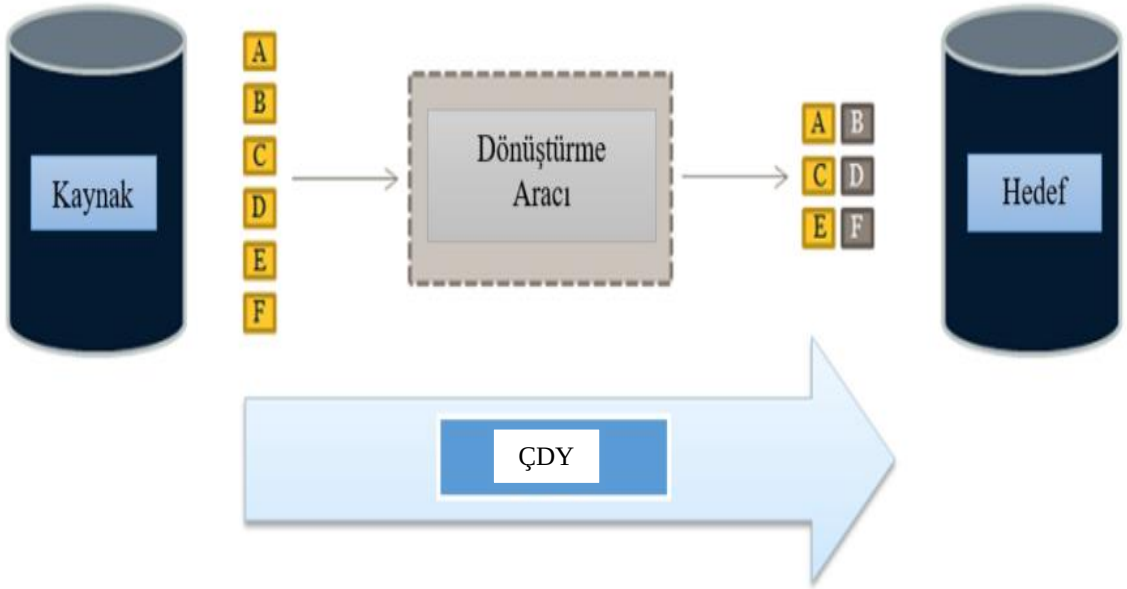
kullanıcı, hedeflediği bilgiye hızlı ve kolayca erişebilir veya sistem yöneticisi, raporlardaki problemlerinin kaynağını hızlıca tespit ederek kısa sürede aksiyon alabilir.

4. VERİ AKTARIM YÖNTEMLERİ

Daha önce de belirtildiği üzere veriler kaynak sistemlerden çıkarılıp hedeflenen ortamlara 2 farklı yöntemle aktarılmaktadır. Bunlar Çek Dönüştür Yükle (ÇDY) ve Çek Yükle Dönüştür (ÇYD) yöntemleridir.

4.1. Çek Dönüştür Yükle (ÇDY)

ÇDY, verilerin kaynak sistemlerden çıkartılması, çeşitli algoritmalara göre dönüştürülmesi ve veri ambarına yüklenmesi işlemine denir. Bu kapsamda, Şekil 4.1'de görüldüğü üzere öncelikle kaynak sistemde yer alan veriler doğru bir şekilde analiz edilir ve ihtiyaca duyulan verilerin çıkartılması işlemi gerçekleştirilir. Farklı kaynak sistemlerden çıkartılmış olan veriler, standart bir formatta bütünleştirilir. Daha sonrasında, iş kurallarına göre belirlenmiş olan veri temizliği, dönüşümü gibi işlemler tamamlanır. Son halini almış olan veri, veri ambarına yüklenir.



Şekil 4.1. ÇDY tasarımı [19]

Veri ambarının efektif çalışması ve verinin kalitesi, direk olarak ÇDY işlemlerinin verimli olmasına bağlıdır. Bu sebeple kaliteli bir ÇDY süreci demek güçlü bir karar destek sistemi demektir. ÇDY sürecine temel teşkil eden geleneksel yaklaşım ise doğru verinin doğru zamanda güncellenmesidir [5].

4.1.1. Verinin çıkartılması

Verinin doğru bir şekilde farklı kaynaklardan çıkartılması ÇDY'nin en zorlu görevlerinden birisidir. Genel olarak çıkartma fazının temel amacı verinin dönüşüm öncesi uygun formatta hazırlanmasıdır. Farklı sistemler farklı formatlarda verilere sahip olabilmektedirler. En çok kullanılan veri formatları düz metin, Excel, XML, ilişkisel veri tabanları ve ilişkisel olmayan veri tabanlarıdır. Çıkarma işlemi sadece kaynak sistemlerden verinin aktarılması sırasında gerçekleşmez. Aynı zamanda, veri ambarı içerisinde yer alan depolama alanlarında tutulan verilerin tablolara aktarımı veya tablolarda tutulan verilerin özet tablolara aktarımı sırasında da çıkarma işlemi uygulanmaktadır.

Genelde kaynak sistemlerde yer alan veriler oldukça karmaşıktır, bu sebeple ilgili olan verinin tespiti zor olabilmektedir. Çıkarma işleminin dizayn edilmesi ve geliştirilmesi fazla vakit isteyen bir aşamadır. Veri ambarının güncellenmesi için veri çıkartma işleminin belirli bir mantığa göre belirli aralıklarla gerçekleşmesi gerekmektedir. Bu noktada da karşımıza 2 farklı veri çıkarma yöntemi çıkmaktadır [5].

- Tam çıkarma: Kaynak sistemde yer alan verinin tamamı tek seferde çıkartılarak veri ambarına aktarımı gerçekleştirilir. Kaynak sistemde gerçekleşen veri değişimi takip edilmez. Genel olarak küçük boyutlu tablolarda bu yöntem tercih edilmektedir.
- Kademeli çıkarma: Bir önceki veri çıkarma işleminden sonra kaynak sistemde gerçekleşen veri güncellemesi, eklenmesi, silinmesi gibi değişiklikler kademeli olarak veri ambarına iletilir. ÇDY uygulamaları, kaynak sistemleri devamlı izleyerek güncellemeleri tespit edebilmekte ve buna göre değişimleri veri ambarına yansıtabilmektedir. İşlem hareket, sıralı kayıt, çağrı kayıt, tahsilat gibi bilgilerin tutulduğu büyük boyutlu tablolarda kademeli çıkarma yapılması hem ÇDY performansı hem de kaynak sistem performansı açısından avantajlıdır.

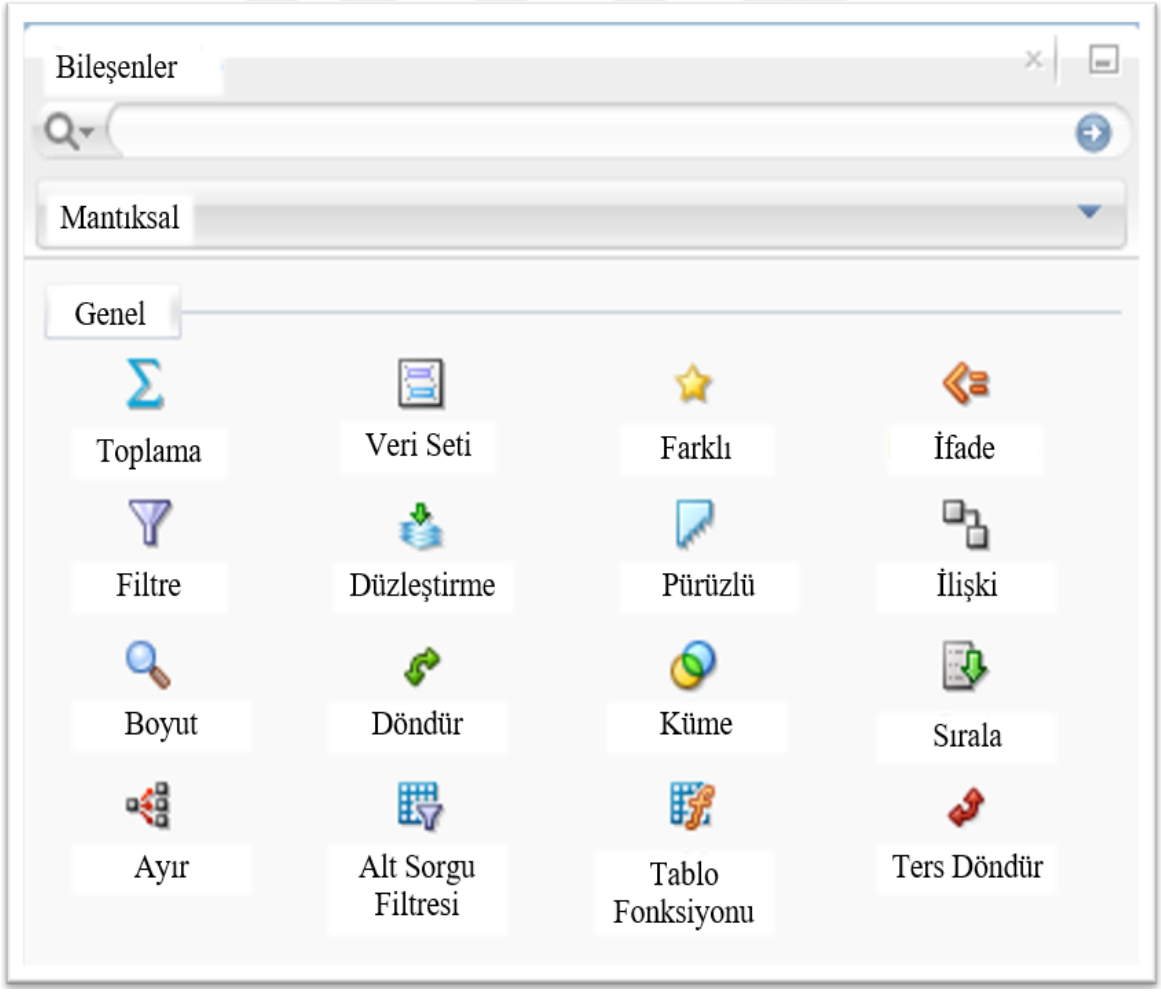
4.1.2. Verinin dönüştürülmesi

Çıkartılmış olan veriye bir dizi kurallar uygulanarak verinin dönüştürülmesi sağlanır. Bu dönüşüm aynı zamanda verinin temizlenmesini de içermektedir. Mesela; cinsiyet bir sistemde kadın-erkek olarak tanımlanmışken başka bir sistemde K-E veya bay-bayan

şeklinde tanımlanmış olabilir. Bunun gibi farklılıklar standart bir formata (Kadın-Erkek) dönüştürülerek giderilmiş olur.

En çok karşılaşılan farklılıklardan birisi de hücre içerisindeki boşluklar ve harflerin küçük veya büyük yazılmasıdır. Bir sistemde şehir ismi "Ankara" gibi yazılırken başka bir sistemde de "ANKARA " gibi yazılabilmektedir. Bu gibi durumlarda da yazımlar ortak bir şekle dönüştürülür ve hücre içerisindeki boşluklar kırpılarak veri temizliği tamamlanmış olur.

Verinin dönüşümü sırasında uygulanan birtakım işlemler Resim 4.1'de gösterilmektedir. Veri setlerinin belirlenmesi, tekil verilerin seçilmesi, yeni tanımlamaların yapılması, çeşitli matematiksel fonksiyonların uygulanması, gereksiz verilerin filtrelenmesi, aktarım sırasındaki tablolar arasındaki ilişkilerin tanımlanması, verilerin sıralanması gibi işlemler veri dönüştürmeye örnek gösterilebilir.



Resim 4.1. ÇDY uygulaması ara yüzü

4.1.3. Verinin yüklenmesi

Dönüşümü tamamlanan veri, veri ambarına yüklenir. Çıkartılma sırasında uygulanan yönteme göre veri yüklemesi yapılır. Kademeli çıkartılan veri kademeli yüklenirken, tam çıkartılan veri de tam olarak yüklenir. Daha önce de belirtildiği gibi, ihtiyaca göre çıkarma ve yükleme işleminin kademeli veya tam olacağına karar verilir. Benzer şekilde veri yükleme işlemleri belirli periyotlar halinde gerçekleşir. Böylece, gereksiz veri yüklemelerinin önüne geçilerek veri ambarının performansı arttırılır ve kullanıcıların ihtiyaç duydukları veriye zamanında erişmeleri sağlanır.

Veri yükleme işlemi veri tabanında yapıldığı için, veri tabanı şemalarında tanımlanacak olan kısıtlar ÇDY işleminin kalite performansını arttırmaya yardımcı olabilir [5]. Mesela, belirli kolonların boş olmaması veya tekil değerleri içermesi gibi genel kurallar kısıt olarak tanımlanırsa, bu koşullara uymayan veriler veri tabanına yüklenmeyecektir.

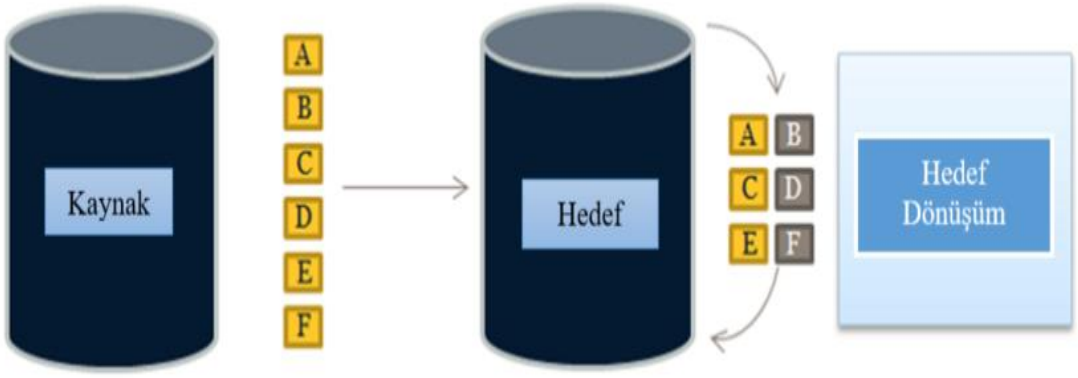
Genel olarak 4 farklı veri yükleme yönteminden söz edilebilir. Bunlar:

- SQL yükleyici: genellikle düz dosyaların veri ambarına yüklenmesinden kullanılır.
- Harici tablolar: harici veriler bu yöntemle sanal bir tablo olarak tanıtılır ve diğer tablolarla ilişkilendirilerek hedef sisteme aktarılır.
- Oracle Çağrı Ara Yüzü (OCI) ve Uygulama Programlama Ara Yüzü (API): veri dönüşümü veri tabanının haricinde yapılacağı zaman kullanılır.
- Çıkartma / Yükleme: herhangi bir dönüşüm işlemine gerek duyulmadığı zaman, veriler olduğu gibi kaynak sistemden çıkarılarak hedef sisteme yüklenir [20].

ÇDY, veri ambarı mimarisinin en önemli kısımlarından birisidir. Çünkü verilerin kaynak sistemden depolama alanına, oradan da hedef sisteme aktarılması ve dönüştürülmesi ÇDY fazında gerçekleşir. Teknolojinin gelişmesi ile birlikte üretilen veri miktarları büyük boyutlara ulaşmakta ve artan rekabet ile birlikte verinin güncelliği her geçen gün daha da önem kazanmaktadır. Firmaların bu tarz ihtiyaçları ancak verimli, hızlı ve performanslı bir ÇDY mimarisi ile karşılanabilir. Bu sebeple, her geçen gün yenilikçi ÇDY yöntemleri önerilerek daha az kaynak kullanımıyla daha hızlı bir veri aktarımı ve daha güncel veri ambarları hedeflenmektedir.

4.2. Çek Yükle Dönüştür (ÇYD)

ÇYD, verilerin kaynak sistemlerden çıkartılması, veri ambarına yüklenmesi, çeşitli algoritmalara göre dönüştürülmesi ve sonrasında hedef tablo veya görsellere yüklenmesi işlemine denir. ÇYD Şekil 4.2'de gösterildiği üzere, verinin kaynak sistemden çıkartılıp veri ambarına aktarılması ve sonrasında yapılacak olan dönüşümlerin veri ambarı üzerinde yapılmasıdır. ÇDY'de verilerin dönüşümü veri ambarına aktarılmadan önce uygulama üzerinde veya sistemde tanımlı olan başka bir veri tabanında yapılmaktadır. ÇYD'de ise dönüşüm veri ambarı üzerinde gerçekleştirilir. ÇYD yönteminde, kaynak sistemdeki veriler birebir veri ambarına kopyalanmaz. Veri ambarında ihtiyaç duyulan veriler kaynak sistemden filtrelenerek çıkartılır ve sonrasında kopyalanır. Böylece kaynak israfı engellenmiş olur.



Şekil 4.2. ÇYD tasarımı [19]

ÇYD yönteminin çıkış noktalarından birisi, veri ambarının sahip olduğu sorgulama gücünün veri dönüşümünde de kullanılması ve böylece ÇDY kaynaklarından tasarruf edilebilmesidir. ÇYD işleminde, depolama alanı için veri ambarında ekstra boş alanlara ihtiyaç vardır.

ÇYD yönteminin avantajları, ayrı bir dönüşüm sistemine ihtiyacın olmaması, veri dönüşümü ve yüklemesinin paralel gerçekleştirilebilmesi, böylece daha kısa sürede daha az kaynağın kullanılması, veri ambarı mimarisinin gelişmiş Hadoop veya bulut mimarisine sahip olması durumunda daha güvenli ve performanslı olmasıdır. Dönüşüm işleminin veri ambarı üzerinde gerçekleşmesi ve farklı veri ambarlarının farklı fonksiyonlara sahip olması geliştiriciler açısından dezavantajlardan birisidir [19]. Bir diğeri ise, veri ambarı üzerinde

gerçekleştirilen dönüşüm işlemleri, kullanıcıların rapor oluşturma ve analiz yapma yeteneklerini olumsuz etkileyebilir.

5. UYGULAMA ÇALIŞMASI

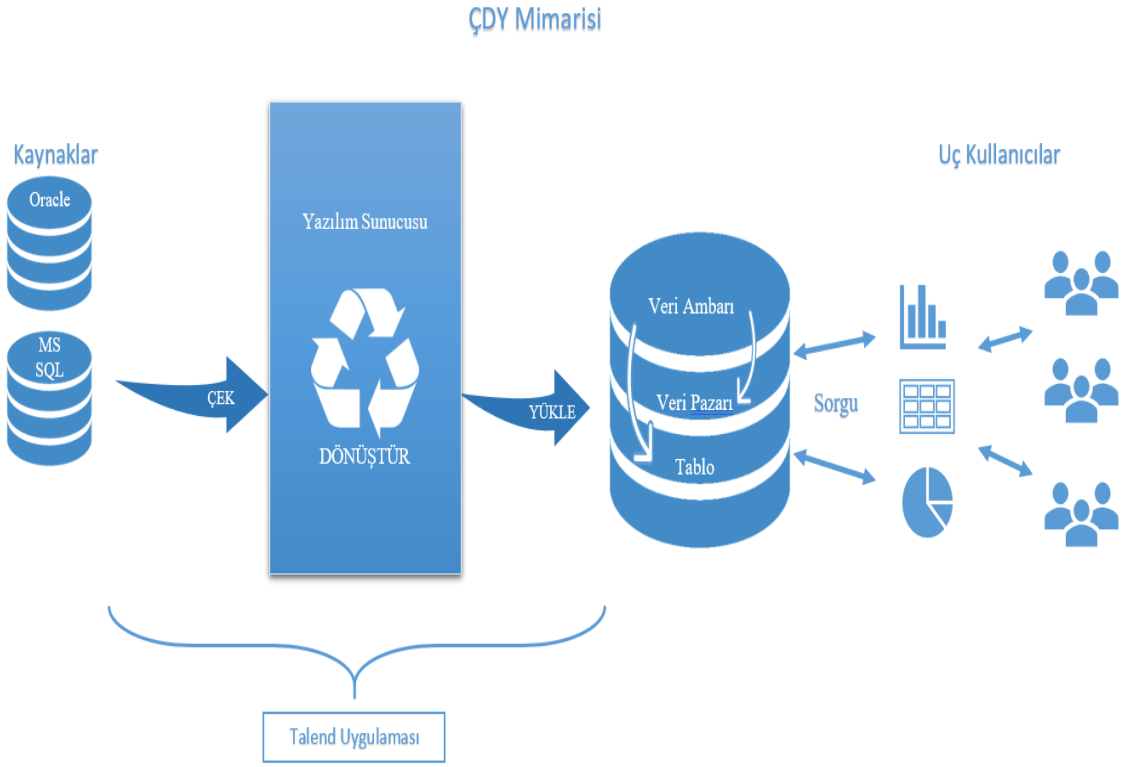
ÇDY ve ÇYD ile ilgili yapılan literatür araştırması sonucunda, güncel veri ambarı mimarisinin oluşturulmasının ortak hedef olduğu gözlemlenmiştir. Bu hedefe ulaşmak için kimi zaman ÇDY yöntemi kimi zaman da ÇYD yöntemi temel alınarak farklı yaklaşımlar geliştirilmiştir. Bu yaklaşımlar bir takım teorik bilgiler ve yapılmış olan başka çalışmalarla desteklenmiştir. Şimdiye kadar, aynı senaryoda, hangi veri aktarımı yöntemin daha iyi olduğu ile ilgili bir bilimsel araştırma yapılmamıştır. Bundan dolayı, yapılmış olan çalışma kapsamında, daha önceden belirlenmiş bir senaryo için ÇDY ve ÇYD yöntemlerinin performans testi yapılarak sonuçları kıyaslanmıştır.

5.1. Uygulama Mimarisi

Performans testinin gerçekleştirilebilmesi için öncelikle bir senaryoya ve senaryonun koşulacağı bir ÇDY/ÇYD mimarisine ihtiyaç vardır. ÇDY için oluşturulmuş olan mimari, Şekil 5.1'de gösterilmektedir.

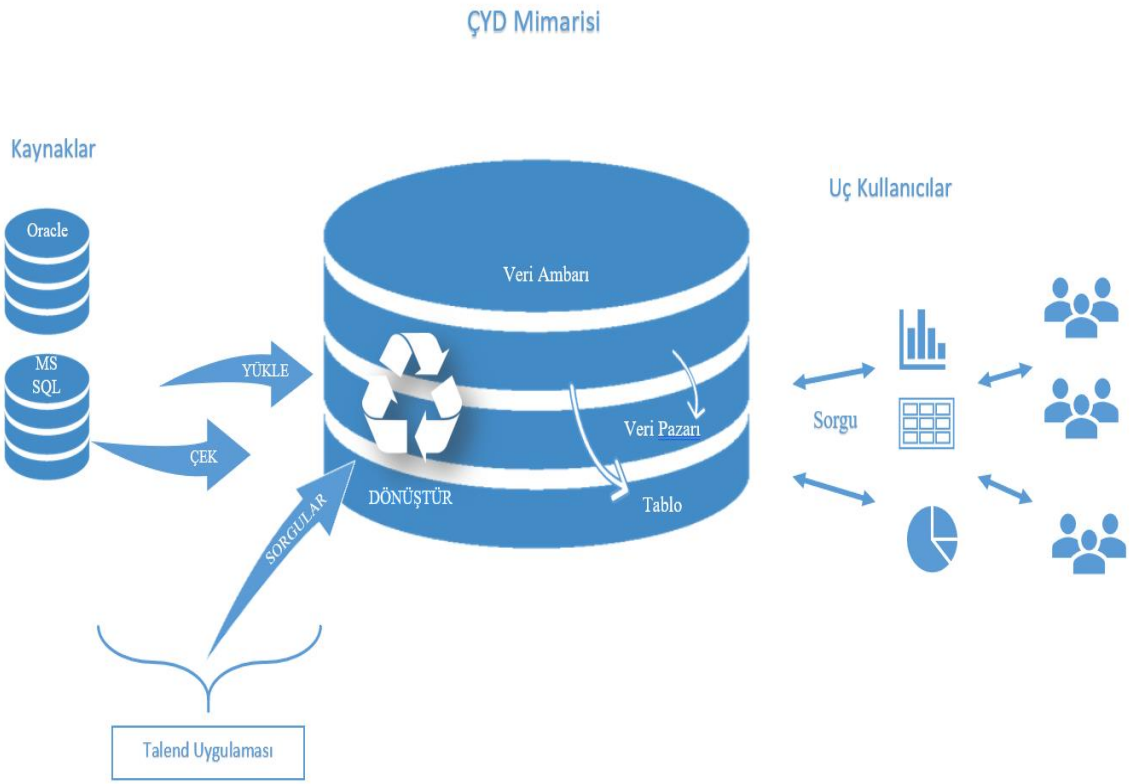
Veri kaynağı olarak Microsoft Developer sitesinde yer alan "School" isimli örnek veri seti kullanılmıştır [21]. Veri modeli hakkında daha detaylı bilgi ayrıca verilecektir. Veri tabanı olarak ta MS SQL Server uygulaması tercih edilmiştir. Çalışma için uygun olan veri aktarma uygulaması olarak Talend tespit edilmiş ve sonrasında gerekli kurulum işlemleri gerçekleştirilmiştir.

Veri ambarı platformu olarak Oracle veri tabanı seçilmiştir. Ayrıca Talend uygulamasında yer alan ELT yöntemi ile veri aktarma özelliği Oracle ile de uyumludur. Oracle veri tabanı hem veri kaynağı olarak hem de veri ambarı olarak kullanılmıştır. Bu sebeple günlük veri işlemleri, veri aktarımları ve rapor sorgulamaları Oracle üzerinde gerçekleştirilmiştir.



Şekil 5.1. ÇDY performans testi mimarisi

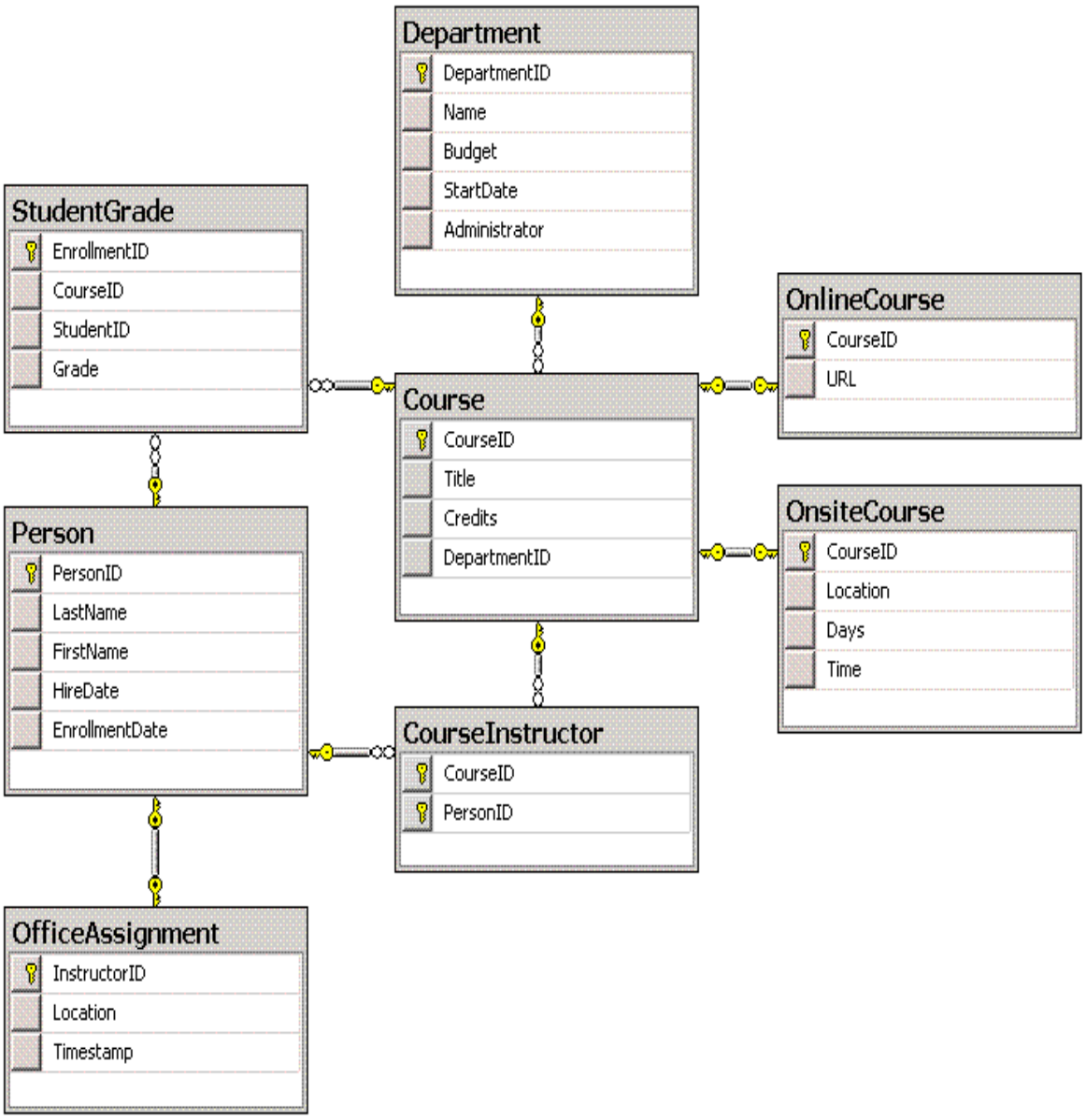
ÇYD için oluşturulmuş olan mimari de Şekil 5.2'de görselleştirilmiştir. Karşılaştırma sonuçlarının mantıklı olması ve gerçeği yansıtabilmesi için ÇDY/ÇYD mimarilerinin birbirlerine benzer olması ve aynı platformda koşturulması hedeflenmiştir. Her iki mimaride de Talend uygulaması kullanılmış, MS SQL ve Oracle ile ilişkiler tanımlanmış ve aynı dönüşümler gerçekleştirilmiştir. Son kullanıcıların veri işleme ve rapor sorgulama periyotları da aynı olacak şekilde planlanmıştır. ÇDY senaryosunda veri dönüşümü, uygulamanın üzerinde gerçekleştirilirken, ÇYD senaryosunda veri dönüşümü veri ambarı üzerinde gerçekleştirilmiştir.



Şekil 5.2. ÇYD performans testi mimarisi

5.1.1. Veri modeli

Daha önce de belirtildiği gibi veri kaynağı olarak Microsoft Developer sitesinde yer alan "School" isimli örnek veri tabanı kullanılmıştır [21]. Veri modeli, basit bir okul bilgi işlem sistemi olarak nitelenebilir. Veri tabanında, Şekil 5.3'te görülebileceği üzere, toplamda 8 tane tablo bulunmaktadır. StudentGrade'i modelinin olgu tablosu olarak tanımlanabilir. Department, Course, Person, OfficeAssignment, OnlineCourse ve OnsiteCourse tabloları da boyut tablolarıdır. CourseInstructor ise iki tablo arasında ilişkiyi tanımlayan bir tablodur. Bu tarz tablolarda ilişki dışında herhangi bir bilgi tutulmamaktadır ancak ilişkinin geçmişi bu tarz tablolardan elde edilebilir. Her bir tablonun kendisine ait birincil anahtarı vardır ve ayrıca diğer tablolarla olan ilişkilerinin tanımlanması için de yabancı anahtar alanları vardır. Tabloların kolonlarının tiplerine baktığımız zaman, içerdiği verinin özelliğine göre değişken karakter, tamsayı, tarih, ondalık ve zaman bilgisi veri tipleri kullanılmıştır.



Şekil 5.3. Veri modeli

Tablolarda yer alan satır sayıları Çizelge 5.1'de incelendiği zaman çok yüksek miktarda verinin olmadığı görülebilir. Ancak, veri aktarımlarında veri boyutları performans kıyaslaması açısından önemli bir kıstastır. Bu sebeple olgu tablosunu rastlantısal verilerle doldurarak ve veri aktarımını hedef tabloya birden fazla defa yaparak tablolardaki veri sayısı arttırılarak hedeflenen miktarlara çıkartılmıştır.

Mevcut modelde olgu tablosu StudentGrade olduğu için veri artışı bu tabloda gerçekleştirilmiştir. Bu sebeple bu tablonun içerisindeki veri sayısı sırasıyla 4566, 493128 ve 2013606 ya çıkarılarak veri aktarımları gerçekleştirilmiştir.

Çizelge 5.1. Tablo satır sayıları

Tablo Adı	Satır Sayısı
Course	20
CourseInstructor	16
Department	6
OfficeAssignment	16
OnlineCourse	8
OnsiteCourse	12
Person	68
StudentGrade	150

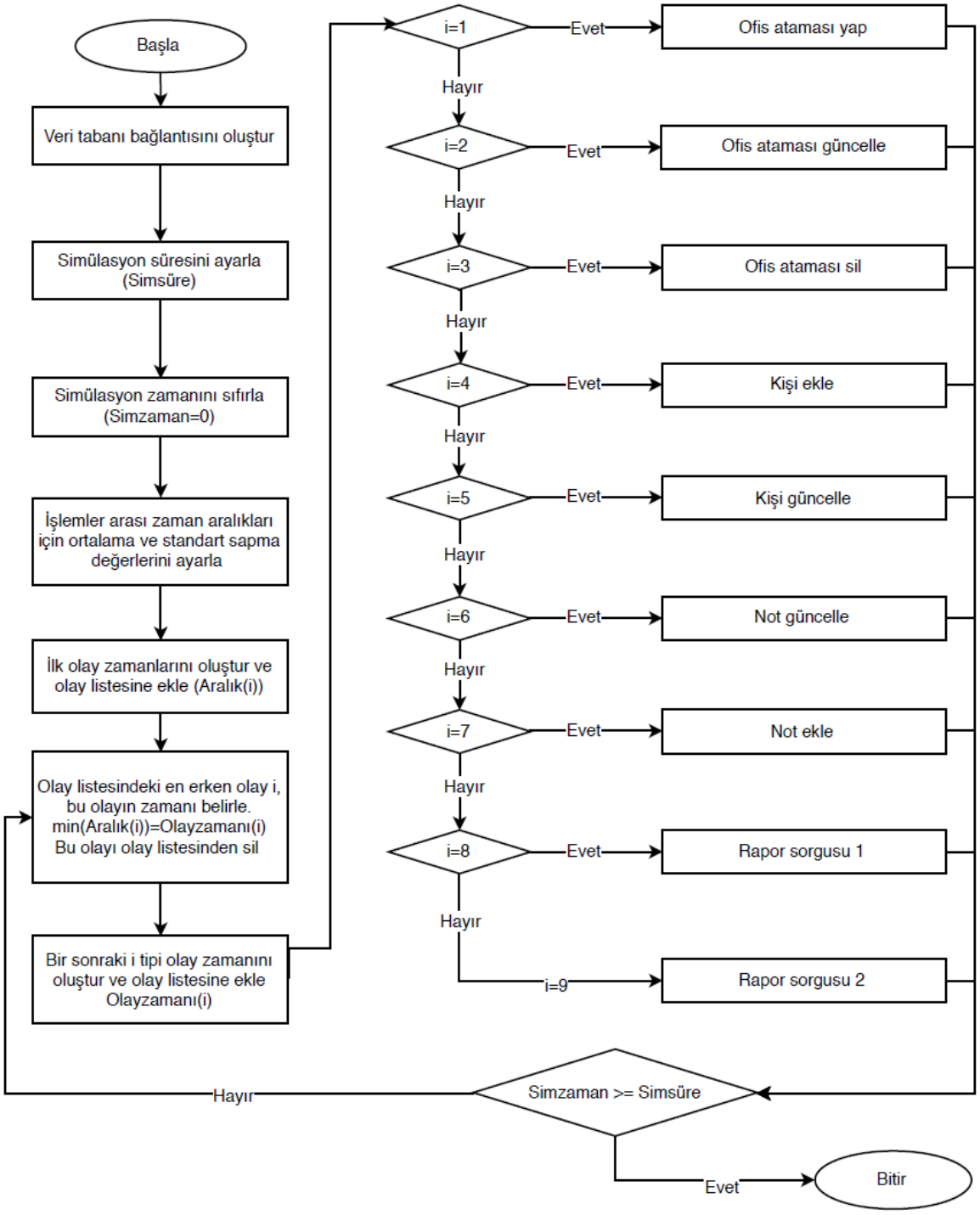
3 farklı veri sayısı elde edilmesindeki amaç, farklı büyüklükteki tabloların veri aktarımı üzerindeki etkisini keşfedebilmektir. Ayrıca sadece tek bir tabloda veri artışı sağlanarak tablo yapısının veri aktarımı üzerindeki etkisi ortadan kaldırılmıştır. Böylece kıyaslamaların daha gerçekçi olması amaçlanmıştır. Veri aktarımı süresini sadece satır sayısıyla ilişkilendirmek yanlıştır. Tabloların kaç kolondan oluştuğu, kolonlardaki veri tipleri de veri aktarımını etkileyen faktörlerdir.

5.1.2. Simülasyon uygulaması

Gerçek hayatta, veri aktarımları anlık, saatlik, günlük, haftalık, aylık veya tek seferlik gerçekleştirilmektedir. Anlık ve saatlik dışında gerçekleştirilen aktarımlar, kullanıcıların veri ambarına bağlanmadığı, sistemlerin boş olduğu gece saatlerinde yapılmaktadır. Anlık veya saatlik olan aktarımlar da mecburen gün içerisinde, kullanıcıların sisteme bağlandığı saatlerde yapılmaktadır. Bu sebeple performans testinin senaryosu 3 farklı koşul altında gerçekleştirilmiştir. İlk koşulda, veri ambarına erişen herhangi bir kullanıcı olmadığı düşünülerek sadece veri aktarımları test edilmiştir. İkinci koşulda ise, tek bir kullanıcının sisteme erişerek veri sorgulaması ve günlük veri işlemlerini yapabildiği düşünülerek test gerçekleştirilmiştir. Son koşulda ise sisteme birden fazla kullanıcının eriştiği düşünülerek veri aktarımı gerçekleştirilmiştir.

Gün içerisindeki yoğunluğu test edebilmek için de Microsoft Visual C#'da bir simülasyon uygulaması geliştirilmiştir. Kullanılan simülasyon yazılımının akış diyagramı Şekil 5.4'te

gösterilmiştir. Daha önce veri tabanında oluşturulmuş olan 7 tane prosedür ve 2 tane rapor sorgusu bu uygulamada olay tipi olarak kaydedilmiştir. Oluşturulmuş prosedürler belirli koşullara göre veri tabanındaki tablolara veri yükleme, tablolarda veri güncelleme, tablolardan veri silme ve veri sorgulama işlemlerini gerçekleştirmektedir. Buradaki amaç, gün içerisinde kullanıcılar tarafından gerçekleştirilen işlemleri belirli aralıklarla veri tabanında gerçekleştirebilmektir. Ayrıca, oluşturulmuş olan sorgular ile de raporların belirli aralıklarla kullanıcılar tarafından görüntülenebildiği uygulamada simüle edilmiştir. Uygulama içerisinde, performans testi için veri tabanı bağlantıları, simülasyon uygulamasının ne kadar süreceği, simülasyonun başlangıç zamanı, 9 farklı olay tipi ve her bir olay tipi için birer ortalama ortaya çıkma süresi (aralık) belirlenmiştir. Aralıklar arası zaman aralığının normal dağılıma uyduğu varsayılmıştır. Dolayısıyla ilgili dağılımların ortalama ve standart sapma süreleri uygulamada belirtilmiştir. Her bir işlemin kendi zamanlarında çalışabilmesi için simülasyon zamanı modülü geliştirilmiştir. Uygulamanın başlamasıyla simülasyon zamanı (simzaman) sıfırdan başlayarak belirlenmiş olan simülasyon süresine (simsüre) kadar ilerleme işlemini gerçekleştirmektedir ve zamanı gelen işlemin olay tipine göre çalışmasını sağlamaktadır. Simülasyon zamanının ilerleme yöntemi kesikli simülasyon zamanı ilerletme yöntemidir. Her bir işlemin çalışması, aynı tipteki bir sonraki yeni işlem zamanını da oluşturmakta ve daha önce oluşturulmuş olan işlemlerle beraber yeniden zamana göre sıralanarak bir sonraki işlemin çalışmasını sağlamaktadır. Simülasyon döngüsü bu şekilde çalışmasına devam etmektedir. Simülasyon zamanı simülasyon süresine eşit veya daha büyük olduğu zaman da simülasyon sona ermektedir.

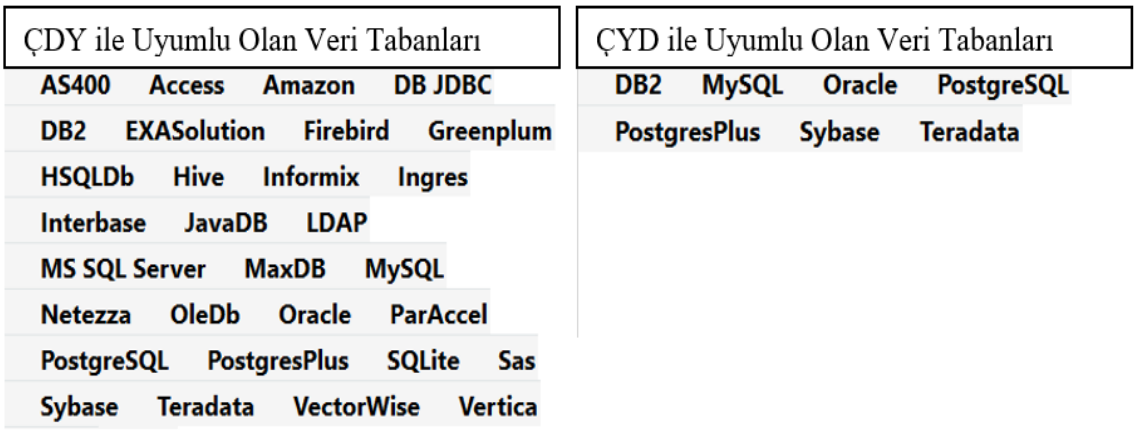


Şekil 5.4. Simülasyon yazılımı akış diyagramı

Tablolara veri yükleyebilme ve tablolardaki verileri güncelleyebilmek için yeni sayı, karakter veya tarih verilerine ihtiyaç vardır. Bu veriler simülasyon yazılımı kapsamında rastgele oluşturulmuştur.

5.1.3. Talend uygulaması

Çalışma için uygun olan veri aktarma uygulaması olarak Talend tespit edilmiş ve sonrasında gerekli kurulum işlemleri gerçekleştirilmiştir. Talend uygulamasının seçilmiş olmasının asıl sebebi hem ÇDY hem de ÇYD yöntemleri ile veri aktarma işleminin yapılabiliyor olmasıdır. Her iki veri aktarma yöntemine sahip olan başka bir uygulama bulunmamaktadır. Diğer uygulamalarda ya ÇDY ya da ÇYD yöntemlerinden biri tercih edilmiştir. Talend, açık kaynak kodlu ücretsiz bir uygulamadır ve birçok veri tabanı ile uyumlu olarak çalışabilmektedir. Şekil 5.5'te Talend uygulamasının veri aktarma yöntemlerine göre uyumlu olduğu veri tabanlarının listesi görülmektedir. ÇDY yönteminin uygulanabildiği veri tabanı sayısı oldukça fazladır. Ancak Talend uygulamasında yer alan ÇYD yöntemi ile veri aktarma özelliği bütün veri tabanları ile uyumlu değildir. Bu sebeple uygulamada veri ambarı platformu olarak Talend'ın uyumlu olduğu Oracle tercih edilmiştir.

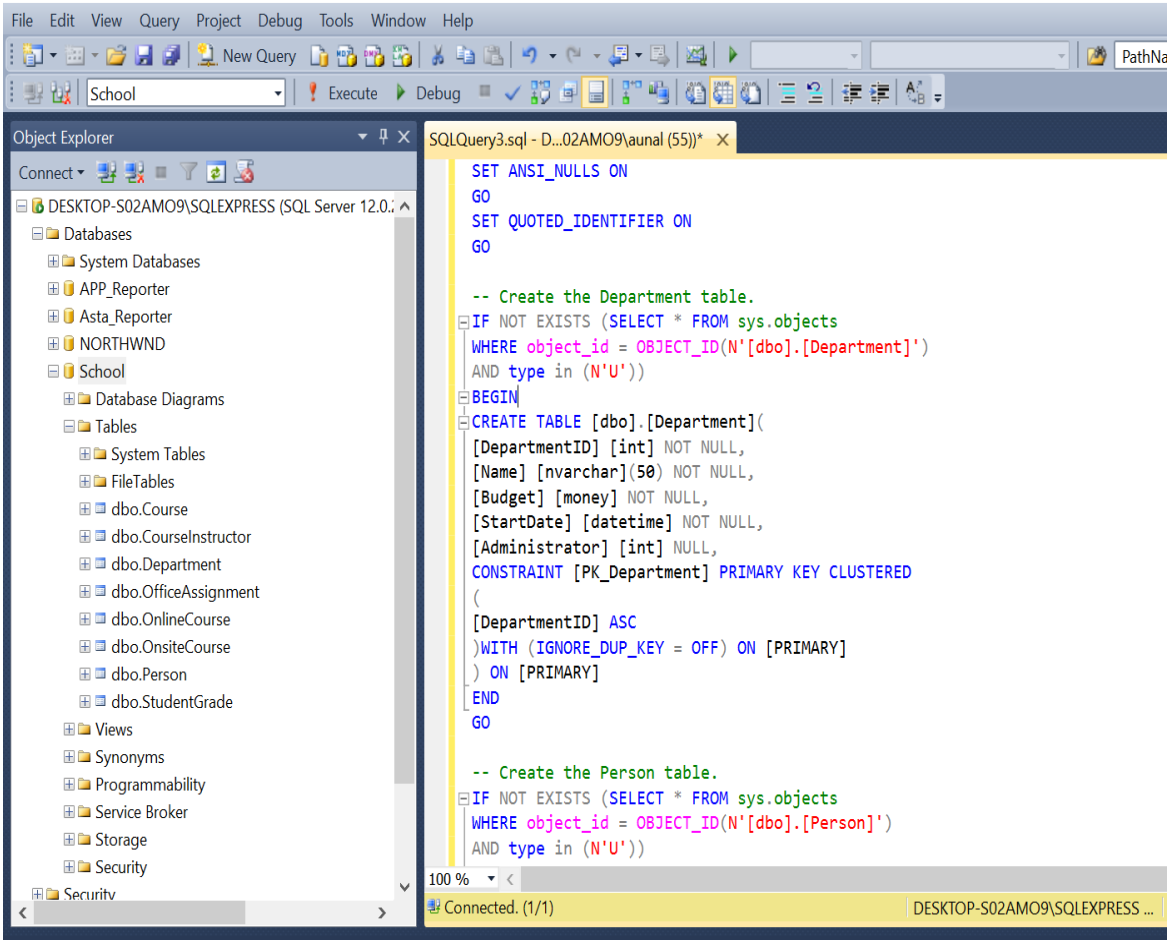


Şekil 5.5. Talend uygulamasının yöntemlere göre uyumlu olduğu veri tabanları

5.2. Uygulamanın Gerçekleştirilmesi

Bu mimarinin oluşturulması için Gazi Üniversitesi Endüstri Mühendisliği laboratuvarında yer alan iş istasyonu kullanılmıştır. İş istasyonunda 2 tane Intel Xeon E5-2640V4 2.40 GHz işlemci, 64 GB yüklü bellek ve 64 bit işletim sistemi yer almaktadır. Bütün uygulamaların kurulumu ve senaryo testleri bu cihaz üzerinde gerçekleştirilmiştir. Öncelikli olarak veri tabanlarının kurulumu gerçekleştirilmiş daha sonrasında sırasıyla Talend ve Microsoft Visual C# uygulamaları yüklenmiştir.

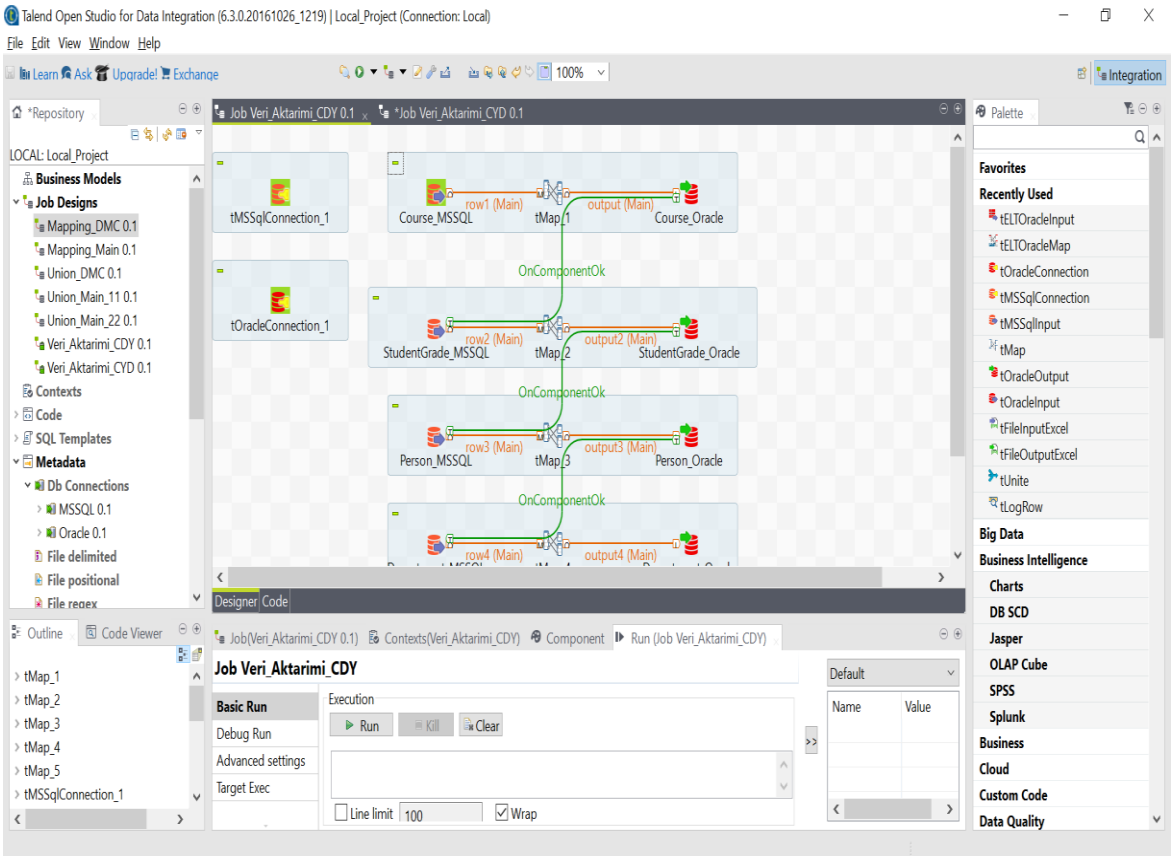
Daha önce de belirtildiği gibi veri kaynağı olarak Microsoft Developer sitesinde yer alan "School" isimli örnek veri tabanı kullanılmıştır [21]. Tabloların, ilişkilerin, prosedürlerin ve örnek verilerin oluşturulmasına yardımcı olan sorgular sitede yer almaktadır. Bu sorguların çalıştırılabilmesi için öncelikli olarak MS SQL Server veri tabanının kurulumu gerçekleştirilmiş ve SQL Server Studio ara yüzü yüklenerek veri tabanına erişim sağlanmıştır. Daha sonrasında Resim 5.1'de görüldüğü üzere yeni bir kullanıcı tanımlanarak, veri tabanı, tablolar oluşturulmuş ve örnek veriler tablolara yüklenmiştir.



Resim 5.1. SQL Server Studio arayüzü

Daha öncede belirtildiği gibi veri ambarı platformu ve veri kaynağı olarak Oracle veri tabanı seçilmiştir. Bu sebeple günlük veri işlemleri, veri aktarımları ve rapor sorgulamaları Oracle üzerinde gerçekleştirilmiştir. Oracle'ın sitesinde yer alan yükleme dosyaları indirilmiş ve veri tabanının kurulumu gerçekleştirilmiştir. Veri tabanına bağlanabilmek için Toad yazılımı yüklenmiş ve bu ara yüzle veri tabanına erişim sağlanmıştır.

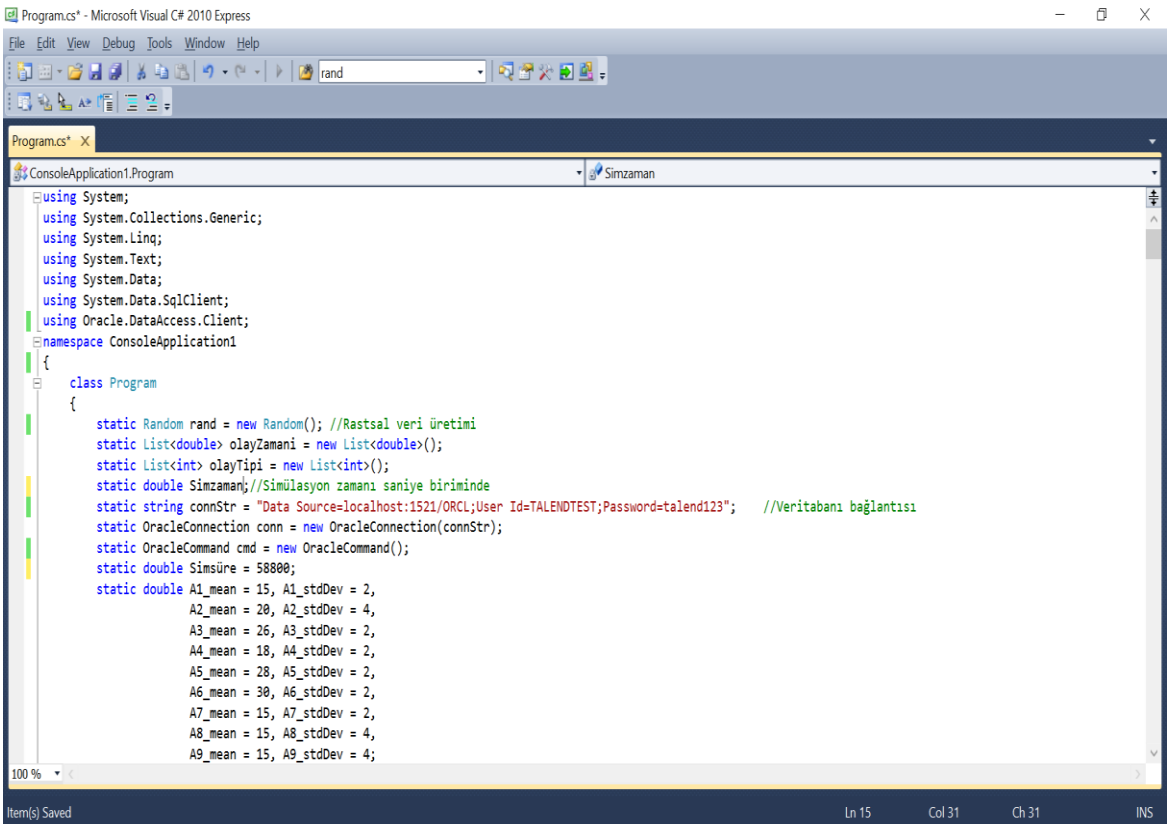
Talend uygulamasının iş istasyonuna yüklenmesi gerçekleştirilmiştir. Resim 5.2'de görüldüğü üzere veri tabanlarının bağlantıları tanımlanmış, veri aktarımı yapılacak olan her bir tablo Talend uygulamasında belirtilmiş ve kolonlar birbirleriyle eşleştirilmiştir. Veri aktarımı yöntemine göre çeşitli iş modelleri geliştirilmiş ve veri aktarımı özellikleri belirtilmiştir. Çalışma kapsamında kullanılan modelde 2 farklı veri aktarımı yöntemi test edileceğinden dolayı ÇDY ve ÇYD için ayrı ayrı iş modelleri oluşturulmuştur. Ayrıca veri çıkarma olarak tam çıkarma tercih edilmiştir.



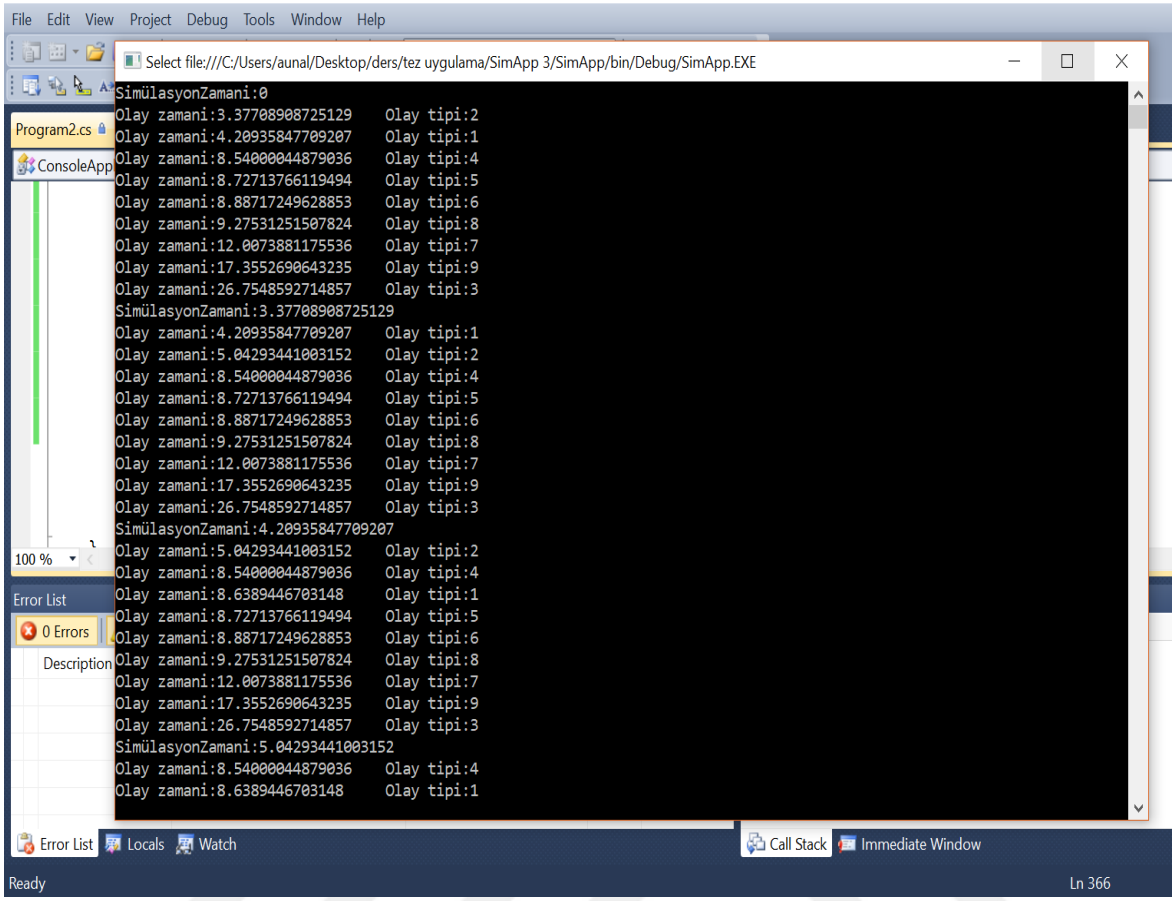
Resim 5.2. Talend uygulaması arayüzü

Simülasyon uygulamasının yazılabilmesi için Microsoft Visual C# uygulaması yüklenmiştir. Uygulama içerisinde Oracle veri ambarına bağlantı sağlayabilmek için ayrıca Oracle Data Access Components (ODAC) 12c uygulaması yüklenmiştir. Bu uygulama olmadan, C#'dan Oracle'a bağlantı sağlanamamaktadır. Simülasyon yazılımı, akış diyagramında belirtilen aşamaların C#'da kodlanmasıyla çalışır hale getirilmiştir. Resim 5.3'te simülasyon uygulamasında yer alan kodların bir kısmı gösterilmiştir.

Simülasyon uygulamanın çalışır durumdaki ekran görüntüsü ise Resim 5.4'te gösterilmiştir. Her bir olay tipi için işlem zamanı oluşturulmakta ve en yakından en uzağa göre sıralanmaktadır. En başta sıfır olan simülasyon zamanı, bir sonraki döngüde en yakın zamanlı olay tipinin zamanına gelmektedir. Aynı tipteki bir sonraki yeni işlem zamanı da oluşturulmakta ve daha önce oluşturulmuş olan işlemlerle beraber yeniden zamana göre sıralanarak bir sonraki işlemin çalışmasını sağlanmaktadır. Simülasyon döngüsü bu şekilde çalışmasına devam etmektedir.



Resim 5.3. Simülasyon uygulaması kod arayüzü



Resim 5.4. Simülasyon uygulamasının çalışması

Bütün yükleme ve yazılım geliştirme işlemleri tamamlandıktan sonra daha önce belirtilmiş olan ÇDY ve ÇYD mimarilerine göre performans testleri gerçekleştirilmiştir ve sonuçları analiz edilmiştir.

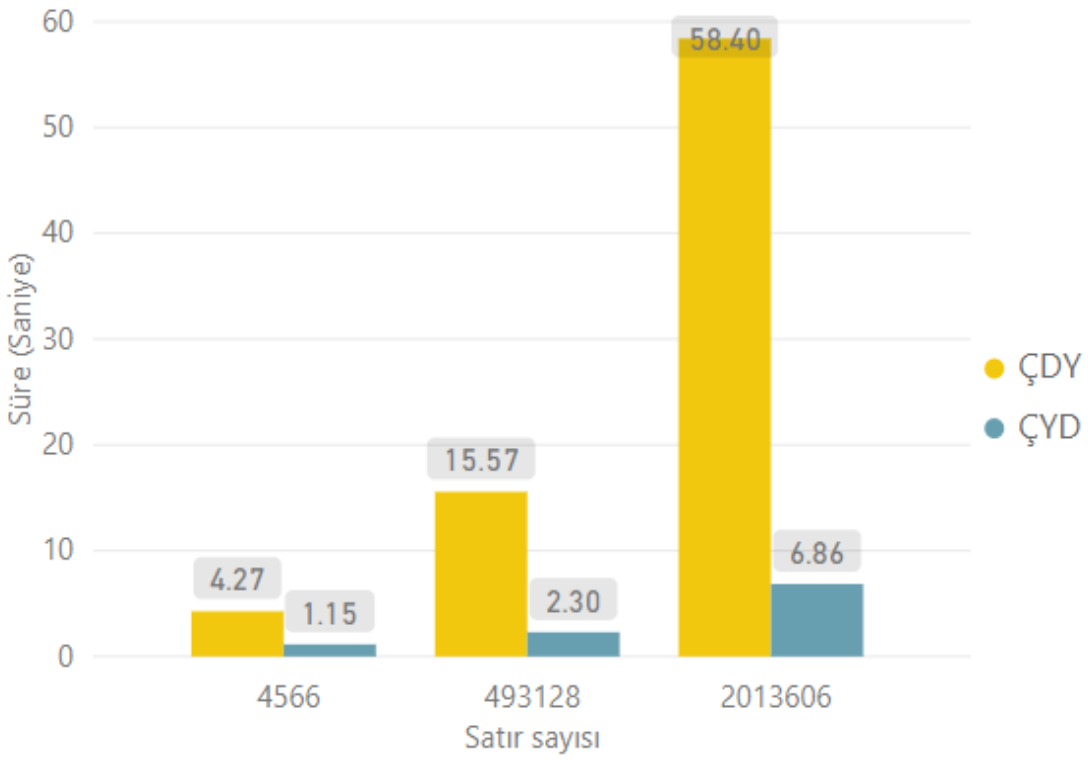
5.3. Sonuçların Analizi

Yapılmış olan testlerin sonuçları, performans ölçütlerine göre değerlendirilip kıyaslanmıştır. ÇDY/ÇYD performanslarının kolay bir şekilde yorumlanabilmesi için, anlaşılması kolay ve basit kriterlerin belirlenip kıyaslamaların yapılması gerekmektedir [8]. Bu bağlamda belirlenmiş olan performans ölçütleri şu şekilde sıralanabilir:

- Veri aktarımı süresi: Veri aktarımı süresinin satır/hücre bazında ortalama değerlerinin hesaplanması.
- Verinin kalitesi: Çıkarımı, dönüşümü ve aktarımı tamamlanan verinin kalitesinin tespit edilerek kıyaslanması.

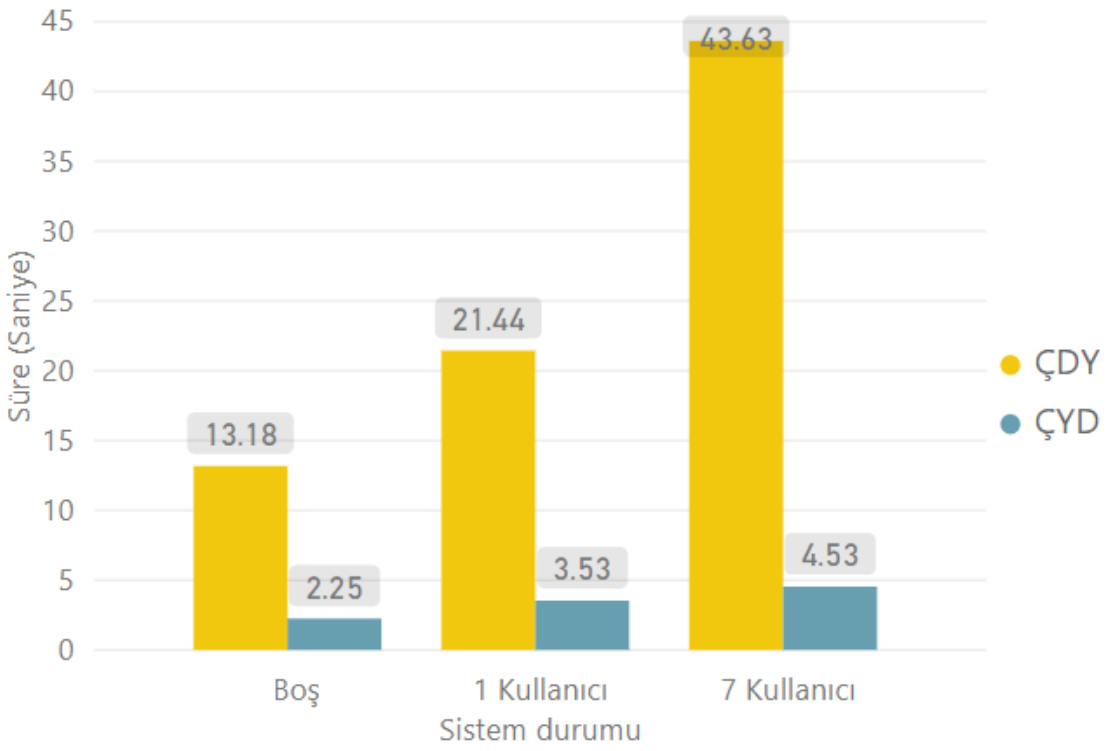
Performans testi mimarisi kısmında bahsedildiği gibi, veri aktarımları 3 farklı koşulda gerçekleştirilmiştir ve böylece ÇDY/ÇYD yöntemleri arasındaki farkı ve ayrıca her iki aktarım yönteminin sistem yoğunluğuna verecekleri tepkiler analiz edilmiştir. Sistem yoğunluğu simülasyon uygulamasının çalıştırılmasıyla oluşturulmuştur. Uygulamanın bir sefer açılarak rastlantısal olarak veri tabanına işlemler (transactions) başlatan komutlar göndermesi tek bir oturum olarak düşünülmüştür. Günlük hayatta kullanılan sistemlere genellikle paralelde birden fazla oturum açılmakta ve böylece yoğunluk arttırılmaktadır. Bunu da test edebilmek için, simülasyon uygulaması birden fazla açılarak paralel oturumlar oluşturulmuştur. Her bir veri aktarımı her bir senaryo için toplamda 15 sefer tekrar edilmiştir, böylece istatistiksel olarak daha doğru bir ortalama değer tespit edilmeye çalışılmıştır.

Daha önce de belirtildiği gibi, satır sayısı veri aktarımı için önemli bir kıstastır. Bu sebeple, veri modelinde bulunan tablolar rastlantısal verilerle doldurarak farklı boyutlarda tablolar elde edilmiş ve veri aktarımları bu tablolar üzerinden test edilmiştir. Tabloların satır sayıları sırasıyla 4566, 493128 ve 2013606 olmuştur. Farklı büyüklükteki tabloların her iki yönteme göre aktarım süreleri Şekil 5.6'da gösterilmektedir.



Şekil 5.6. Satır sayısına göre ÇYD ve ÇDY ortalama süreleri

Çıkan sonuçlar analiz edildiğinde ÇYD yönteminin ortalama aktarım süresi 3 farklı boyuttaki tablo için daha performanslı çıkmıştır. ÇDY yönteminin ise daha yavaş olduğu görülmektedir. Küçük boyuttaki tabloda ÇYD yöntemi ÇDY'den 3,7 kat daha hızlı iken yüksek boyuttaki tabloda bu fark 8,5'e çıkmıştır. Buna göre, veri boyutu arttıkça ÇYD yönteminin performansının arttığı söylenebilir.

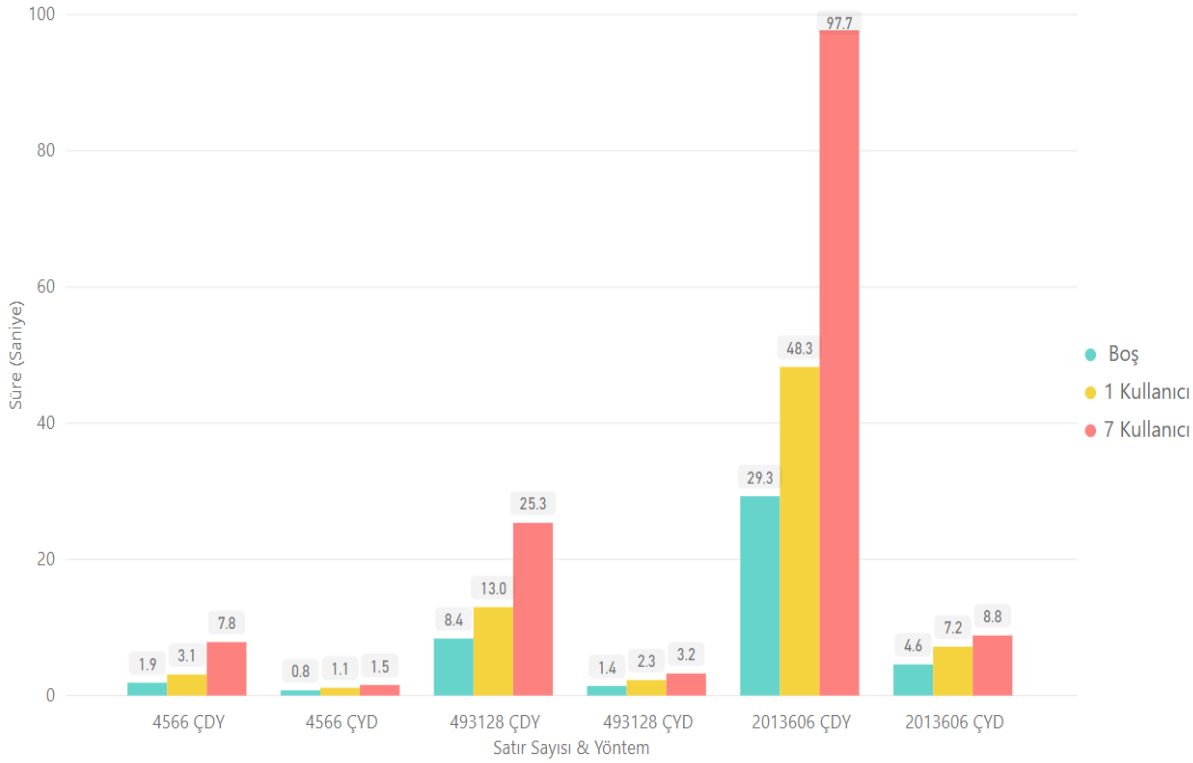


Şekil 5.7. Sistem yoğunluğuna göre ÇYD ve ÇDY ortalama süreleri

Sonuçlara bir de sistem yoğunluğu açısından bakıldığında, benzer şekilde ÇYD'nin ÇDY yönteminden daha performanslı olduğu Şekil 5.7'de görülmektedir. Sistemde kullanıcı yokken ÇYD yaklaşık 6 kat daha hızlı iken sistem yoğunlaştığında bu oran yaklaşık 10 katına çıkmaktadır. Bu doğrultuda, sistem yoğunluğu arttıkça ÇYD yönteminin performansının arttığı söylenebilir.

Sonuçlara sistem yoğunluğu açısından daha detaylı olarak bakıldığında, ortalama aktarım süreleri Şekil 5.8'deki gibi değişmektedir. Farklı boyutlardaki tabloların ve veri aktarımı yöntemlerinin sistem yoğunluğuna vermiş oldukları tepki açık bir şekilde görülebilmektedir. Küçük boyuttaki tabloların aktarım süresi sistem yoğunlaştıkça ÇYD yönteminde yaklaşık

2 kat artarken ÇDY yönteminde yaklaşık 4 kat artmaktadır. Orta ve büyük ölçekli tablolarda da benzer bir değişim gözlenmektedir.



Şekil 5.8. Satır sayısı ve yönteme göre sistem yoğunluğunun ortalama süreleri

ÇDY ve ÇYD yöntemleri, veri kalitesi açısından incelendiğinde bir fark tespit edilememiştir. Aktarımı yapılmış olan tablolarda eksik veya kaynaktan farklı bir veriye rastlanılmamıştır. Bu noktada, modelin uygulamada doğru olarak kurgulanması veri kalitesi açısından önemli olmaktadır. Veri aktarımı başlatılmadan önce, uygulama veri kalitesinde problem olmaması için gerekli ön analizleri yapmakta ve kullanıcıya uyarı vermektedir. Örneğin, kaynakta bir alan karakter olarak tanımlanmışken hedefte sayısal bir değer olarak tanımlanırsa, uygulama bunu tespit edip uyarısını vermekte ve aktarımı başlatmamaktadır. Veya kaynakta sayısal olan kolonun bazı satırlarında karakter veriler varsa bunlar hedefe sayısal olarak aktarılmamaktadır. Daha önce de belirtildiği gibi veri aktarımı uygulamaları sadece veri aktarımını yapmamakta aynı zamanda veri dönüşümü ve veri kalitesini de sağlamaktadır.

Talend uygulamasında model oluşturulurken ÇDY yönteminin ÇYD yöntemine göre daha fazla veri tabanıyla uyumlu olduğu Şekil 5.5'te gösterilmiştir. Yapılmış olan test sonuçlarına göre ÇYD yöntemi daha performanslı gözükmektedir ancak her veri tabanı sisteminde bu

yöntem uygulanamamaktadır. Talend uygulamasındaki ÇYD yöntemi Oracle, MS SQL, Sybase, Teradata gibi veri tabanlarıyla uyumludur. Bu veri tabanları daha çok veri ambarı platformu olarak tercih edilmektedir.

6. SONUÇ VE ÖNERİLER

Bu çalışma kapsamında, teknolojinin gelişmesiyle birlikte veri ambarı ve iş zekâsı ürünlerine duyulan ihtiyacın artmasından bahsedilmiştir. Veri ambarını oluşturan katmanlar tanımlanmış ve veri aktarımının bu katmanlar içerisindeki önemi vurgulanmıştır. Yapılmış olan literatür araştırmaları sonucunda veri aktarımı için 2 farklı temel yöntemin kullanıldığı tespit edilmiş ve bu yöntemlerin (ÇDY ve ÇYD) detaylı tanımlamaları yapılmıştır. Yapılmış olan çalışmada, bu iki yöntem belirli senaryolara göre test edilerek sonuçları değerlendirilmiştir.

Performans testi, simülasyon uygulaması aracılığıyla oluşturulan 3 farklı koşul altında gerçekleştirilmiştir. Çıkan sonuçlar genel olarak analiz edildiğinde ÇYD yönteminin ÇDY yönteminden daha hızlı ve performanslı olduğu tespit edilmiştir. Buna karşılık ÇDY yönteminin de daha fazla veri tabanı sistemiyle uyumlu olduğu gözlemlenmiştir. Talend uygulamasında ÇYD yönteminin uygulanabildiği veri tabanı sayısı kısıtlıdır.

Sonuç olarak, farklı boyutlardaki veriler ve farklı yoğunluktaki sistemler için daha güncel bir veri ambarına ihtiyaç duyulduğunda, performans açısından, ÇYD yönteminin ÇDY yöntemine tercih edilebileceği değerlendirilmiştir.

KAYNAKLAR

1. İnternet: Gündüz, D. (2013). Business intelligence (iş zekâsı)'e giriş. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fderyagunduz.com%2F%3Fp%3D805&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

2. İnternet: SINTEF. (2013). Big Data, for better or worse: 90% of world's data generated over last two years. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.sciencedaily.com%2Freleases%2F2013%2F05%2F130522085217.htm&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

3. Simitsis, A., Vassiliadis P., Sellis T. (2005). *Optimizing etl processes in data warehouses*. Proceedings of the 21st International Conference on Data Engineering, IEEE, Tokyo, Japan, 564-575.

4. Vassiliadis, P. (2009). A survey of extract-transform-load technology. *International Journal of Data Warehousing and Mining*, 5(3), 1–27.

5. Kakish, K., Kraft, T. (2012). *ETL evolution for real-time data warehousing.* Proceedings of the Conference on Information Systems Applied Research, New Orleans Louisiana, USA, 2167(1508).

6. Ferreira, N., Pedro, M., Pedro, F. (2013). *Near real-time with traditional data warehouse architectures: factors and how-to.* Proceedings of the 17th International Database Engineering & Applications Symposium, Barcelona, Spain, 68-75.

7. Sharma, S., Jain, R. (2014). *Modeling ETL process in data warehouse: an exploratory study*, in Proceedings of 4th International Conference on Advanced Computing & Communication Technologies (ACCT2014), India, 271-276.

8. Wyatt, L., Caufield, B., Pol, D. (2009). *Performance Evaluation and Benchmarking*. Berlin: Springer-Verlag, 183-198.

9. Freudenreich, T., Furtado, P., Koncilia, C., Thiele, M., Waas, F., Wrembel, R. (2012). *An on-demand ELT architecture for real-time BI*. VLDB Workshop Enabling Real-Time Business Intelligence (BIRTE), 50-59.

10. Waas, F., Wrembel, R., Freudenreich, T., Theile, M., Koncilia, C., Furtado, P. (2013). On-demand ELT architecture for right-time BI: extending the vision. *International Journal on Data Warehousing and Mining*, 9(2).

11. Marin-Ortega, P. M. (2014). ELTA: new approach in designing business intelligence solutions in era of big data. *Procedia Technology*, 667-674.

12. Inmon, W. H. March (2002). *Building the data warehouse*. New York: John Wiley & Sons Inc., 40-45.

13. Kimball, R. February (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses.* New York: John Wiley & Sons Inc., 235-240.

14. Menolli, A. L. A., Dias, M. M. (2006). *A data warehouse architecture in layers for science and technology*. In Proceedings of the Eighteenth International Conference onSoftware Engineering & Knowledge Engineering, SanFrancisco, California, 162-165.

15. İnternet: Demarest, M. (1997). The politics of data warehousing. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.noumenal.com%2Fmarc%2Fdwpoly.html%23note2&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

16. Song, Y., Rowen, W., Medsker, C. ve Ewen, E. (2001). *An analysis of many-to-many relatioships between fact and dimension tables in dimensional modeling.* In: Proceedings of International Workshop on Design and Management of Data Warehouses (DMDW'01), Zurich, Switzerland, 6.1-6.13.

17. İnternet: 1Keydata (2018). Snowflake schema. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.1keydata.com%2Fdatawarehousing%2Fsnowflake-schema.html&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

18. İnternet: Gökyokuş, Ü. (2016). Veri ambarında kurumsal veri modelinin oluşumu. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fwww.komtas.com%2Fblog%2F15%2Fveri-ambarinda-kurumsal-veri-modelinin-olusumu&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

19. İnternet: Speare, G. (2015). ETL vs. ELT – what's the big difference?. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.ironsidegroup.com%2F2015%2F03%2F01%2Fetl-vs-elt-whats-the-big-difference%2F&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

20. İnternet: Oracle (2005). Oracle database data warehousing guide, 10g Release 2 (10.2). URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdocs.oracle.com%2Fcd%2FB19306_01%2Fserver.102%2Fb14223.pdf&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

21. İnternet: Microsoft (2016). Entity framework school database. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fmsdn.microsoft.com%2Fen-us%2Flibrary%2Fjj614587%28v%3Dvs.113%29.aspx&date=2018-06-12, Son Erişim Tarihi: 18.04.2018.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: ÜNAL, Abdullah
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 20.07.1988, Erzurum
Medeni hali	: Evli
Telefon	: 0 (505) 056 24 66
e-mail	: aunal@outlook.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi / Endüstri Mühendisliği	Devam ediyor
Lisans	ODTÜ / Endüstri Mühendisliği	2011
Lise	Atatürk Anadolu Lisesi	2006

İş Deneyimi

Yıl	Yer	Görev
2017-Halen	Think Elevate	İş Zekâsı Uzmanı
2016-2017	ICTerra	İş Zekâsı Uzmanı
2011-2016	Türk Telekomünikasyon A.Ş.	İş Zekâsı Uzmanı

Yabancı Dil

İngilizce

Yayınlar

1. Ünal, A., Çetinyokuş, T., Kellegöz, T. (2017). *Veri Ambarı Oluşturmada ETL ve ELT Yaklaşımlarının Kullanımı ve Karşılaştırılması,* International Advanced Researches & Engineering Congress, Osmaniye.

Hobiler

Gezi, Kitap okuma

GAZİ GELECEKTİR...