

ÇİFT ZAMANLI VE YARI YAPILI ÖZEL VERİ AMBARLARI İLE İLİŞKİSEL VERİ TABANI ÜZERİNE BİR KARŞILAŞTIRMA ÇALIŞMASI

Gözde ALP^{*1}, Canan Eren ATAY²

¹ Marmara Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye

² Dokuz Eylül Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İzmir, Türkiye

Özet: Veri ambarları ve ilişkisel veri tabanları büyük veriyi saklamak ve işlemek için kullanılan iki farklı depolama alanıdır. Bu çalışmada; yapısal olarak özelleştirilmiş iki zamanlı ve iç içe yapıda bir veri ambarı hazırlanmıştır. Aynı veriler veri ambarında ve ilişkisel veri tabanında saklanarak bu iki depolama alanı arasında karşılaştırma yapılmıştır. Aynı sonucu verecek sorgular hem veri tabanı tabloları kullanılarak hem de veri ambarı bileşenleri kullanılarak sorgulanmıştır. Sonuçlar sorgu setleri, yürütme zamanları ve disk alanı kullanımı açısından değerlendirilmiştir ve karşılaştırılmıştır.

Anahtar Kelimeler: Veri ambarları; büyük veri; öznitelik zaman etiketi; çift zamanlı veri tabanları

A COMPARATIVE STUDY ON BITEMPORAL SEMISTRUCTURED DATA WAREHOUSES AND RELATIONAL DATABASES

Özet: Data warehouses and relational databases are two different areas for data storage. A special structured data warehouse is prepared in this study that is bitemporal and its structure is formed as nested. Same data is stored in data warehouse and the relational database. These two data storage area is competed with each other. Same resulting queries are queried by using relational database objects and data warehouse components. The results are compared and evaluated in terms of query sets, processing time and disk space usage.

Keywords: Data warehouse; big data; bitemporal databases; attribute time stamping.

Giriş

Günümüzde teknolojinin etkileri her sektörde görülmektedir. Sağlıktan, tekstile, eğitimden, yiyecek içecek sektörüne kadar her alanda, bilgiler bilgisayar ortamında kayıt altında tutulmakta ve işlenmektedir. Bunun yanı sıra, insan vücudunun bir uzantısı haline gelen akıllı cep telefonları ile insanlar sosyal ortamlarda kişisel bilgilerini, beğenilerini paylaşır hale gelmiştir. Bilgi paylaşımında güvenilir yöntemlerin geliştirilmesiyle ve veri paylaşım hızının artmasıyla, özellikle son yıllarda internet üzerinden eşya ve yiyecek alışverişi dünyada ve ülkemizde insan hayatının sıradan bir parçası haline gelmiştir. Bilgisayarsız bir hayat düşünülemez hale gelmiştir.

Bilgisayar kullanımı bu denli yaygın değilken kayıt altında tutulamayan birçok veri zamanımızda kayıt altında tutulabilmektedir. Ortaya çıkan büyük veri akışı

verileri işleme ve bu verileri firmaların ve toplumların yararına kullanma ihtiyacı ortaya çıkmasına sebep olmuştur. Öyle ki artık şirketler tüketici eğilimlerini anlamak ve karlarını arttırmak adına ellerindeki verileri işleyecek ve akılcı önerilerde bulunabilecek firmalar ile çalışmaktadır. Dahası bazı büyük şirketlerin veri işleme bölümleri bulunmaktadır. Bunun yanında hastalık teşhis ve tedavilerini geliştirebilmek için sağlık alanında da büyük veri topluma faydalı olması için kullanılmaktadır.

Bu ihtiyaç doğrultusunda veri işleme ve veriden bilgi elde edebilme üzerinde çalışan bilimler ortaya çıkmıştır. Karar destek sistemleri, veri madenciliği yöntemleri büyük veriyi anlamlandırmak için ortaya çıkmış bilimsel alanlardır. Veri ambarları ise yüksek boyuttaki veriyi saklamak ve işlemek için elverişli bir depolama alanlarıdır. Şirketlerin büyük bir çoğunluğu veri ambarında sakladıkları verileri karar destek sistemleri

ya da veri madenciliği yöntemleri ile işleyerek ortaya çıkan bilgiyi kullanarak karar almaktadır.

Akla gelen her sektörde ve sosyal platformda ortaya çıkan bu veriler ilişkisel veri tabanlarında ya da veri ambarlarında saklanabilir. Günümüzde; Microsoft, Oracle gibi şirketler tarafından ya da açık kaynak kodlu olarak üretilmiş onlarca değişik ilişkisel veri tabanı ve veri ambarı sistemleri mevcuttur.

İlişkisel veri tabanı; her an veri ekleme, silme, update etme gibi işlemlerin yapılabilirdiği çevrimiçi olarak adlandırabileceğimiz veri depolama alanıdır. Veri ambarında ise çevrimiçi işlem yapılmaz, sadece sorgulama, raporlama işlemleri yapılır. Veri belli zamanlarda ilişkisel veri tabanı ya da benzer veri depolarından veri ambarına aktarılır.

Sahip olunan eski, önceki yıllara dönemlere ait veriler, gelecek dönemlerdeki stratejileri belirlemekte kullanılabilir. Veri ambarları özel yapıları ile geçmiş dönem analizinin yapılmasını kolaylaştırır. Veri ambarı içerisinde bulunan veriler zamansal derinliğe sahiptir. Veri ambarları, büyük miktarda verinin saklanabileceği bir depolama alanı sunarken aynı zamanda çevrimiçi işleminin olmadığı sadece sorgulama işlemleri yapılan ve dolayısıyla daha hızlı sonuç veren de bir ortam sağlar.

Veri ambarı; eski verilere ulaşmak ve analiz etmek amacı ile kullanılan, aktif bir çalışma alanıdır. Veri ambarı “yöneticilerin karar vermelerini kolaylaştıran; konu odaklı, entegre, kalıcı, zamanla değişen verilerin oluşturduğu bir yığındır” (Inmon ve Krishnan, 2011). Veri ambarı özel hazırlanmış verileri hızlı ve kolay bir şekilde elde etmeyi amaçlar. Şirketlerin çeşitli lokasyonlarda bulunan veri depolarından alınan bu özel veriler, yönetim raporlamalarında, çeşitli işyeri sorgularında, karar destek sistemlerinde, yönetici bilgi sistemlerinde ve veri madenciliği uygulamalarında kullanılır (Ponniyah, 2011).

Veri ambarları; doksanlı yılların başlarından beri, karar destek sistemlerinin başarısında önemli rol oynar. Kuşkusuz, veri ambarı kullanıcıları verilerin doğru ve

eksiksiz olduğundan emin olma ihtiyacı duyar. Veri ambarının, geçici olmayan ve zamanla değişen verileri içermeye özelliği, hali hazırda var olan verileri yok etmeden veriler üzerinde değişiklik yapmaya izin vermeyi gerektirir.

OLAP (online analytical processing) çevrimiçi analitik işleme canlı, gerçek ve hazır dataya ulaşma ihtiyacından dolayı ortaya çıkmıştır. OLAP raporları ile istatistiksel güvenilirliğe sahip gelecek durum analizleri yapmak mümkündür. OLAP küplerindeki veriler günün belli zamanlarında güncellenir ve işlenir. Toplam, ortalama hesaplama gibi tüm hesap işlemleri veri değiştiğinde yeniden hesaplanır. Rapor OLAP küpleri ile sunulduğunda arka planda hiçbir hesaplama yapılmaz, bir öndeki veri güncellemesinde hesaplanmış veriler gösterilir. Tüm hesaplanan değerler daha önce olduğu gibi OLAP küplerinde saklanır. Raporlama esnasındaki yegâne işlem, raporu çağırmak ve kullanıcıya göstermektir. Subotić ve arkadaşları (2014) ve Ravat ve Teste (2006) OLAP boyutları üzerinde versiyonlama ve dinamik güncelleme çalışmalarda yapılmıştır. Tipik OLAP işlemleri arasında bir ya da birden çok boyut hiyerarşisi arasında veri toplama seviyesini yükseltme (rollup) ve veri toplama seviyesini azaltma (drill-down), küçük bir parçayı seçip gösterme (slice_and_dice), ve verilerin çok boyutlu görünümünün yeniden yönlendirilmesi (pivot) bulunmaktadır (Ceci ve Ark,2013; Wrembel, 2009).

Zamansal veri tabanı çalışmaları bu alanda önemli sonuçlar ortaya çıkarmıştır. Tarihsel bir veri tabanı ilgili nesne ya da özelliğin geçmiş zamandaki bilgisini tutabileceği gibi (olay zamanı), veri tabanındaki işlem geçmişini de (işlem zamanı) tutabilir. Olay zamanı nesnelerin ya da özelliklerin zamanla değişen durumlarını modellemek için kullanılır. Bu durumlar geçmişte, günümüzde ya da gelecekte olabilir. İşlem zamanı nesne ya da özelliğin veri tabanına kaydolması ve sonrasında yapılan her türlü değişikliğin zaman etiketini belirtir. Günümüzde kullanılan veri tabanlarında işlem zamanı şimdiki zamanın ötesinde

olamaz, geçmiş zaman ya da gelecek zaman bilgilerini içermez. Hâlbuki çift zamanlı veri tabanları değişen dünyanın değişen bilgilerini modelleyebilir, verilerden elde edilmiş gerçeklerin olay zamanlarını ve nesnelere üzerinde yapılan veri tabanı işlemlerinin geçmişini tutabilir (Atay ve Tansel, 2009).

Veri ambarlarında zamansal bilgi depolanır, bununla ilişkili olarak zamansal veri tabanları üzerinde yapılan araştırmalar, veri ambarları açısından da yararlıdır. Zamansal veri ambarları, zamana bağlı değişen çok boyutlu verilerin yönetimini sağlar. Daha önceden de zamansal veri tabanları ve veri ambarlarını bir araya getiren araştırma projeleri yapılmıştır. Malinowski ve Zimanyi (2006) yılında yaptıkları çalışmada, zamana bağlı olarak değişen ve zamana bağlı olarak değişmeyen çok boyutlu bir model sunmuşlardır. Martin ve Abello (2003) veri ambarları ve zamansal veri tabanları arasındaki benzerlik ve farklılıkları defatlı bir şekilde yaptıkları çalışmada sunmuşlardır. Eder ve arkadaşları (2001) çalışmalarında, zamansal veri ambarı modelini iki farklı yaklaşım kullanarak sunmuşlardır. Bunlardan biri olan doğrudan yaklaşımda, her sonucunu elde etmek için bir sorgu analizörü ve sonuç analizörü kullanılır. Bir diğeri doğrudan olmayan yaklaşımda, data mart adı verilen küçük veri ambarları belli yapısal özelliklere göre oluşturulur. Diğer tüm yapı versiyonlarındaki veriler gerekli dönüşüm fonksiyonları ile işlenerek yeni oluşturulan data marta aktarılır. Doğrudan yaklaşım daha fazla esneklik sunarken, dolaylı yaklaşım data mart bir kere oluşturulduktan sonra tepki süresi açısından üstündür. Janet ve arkadaşları (2006) veri ambarı hali hazırda var olan veri ambarı küpleri üzerindeki boyutların değiştirilmesi, silinmesi, küplere yeni boyut eklenmesi gibi işlemleri çalışan uygulamalara zarar vermeden yapan bir model geliştirmişlerdir. Ayrıca çok boyutlu şemalarda; ekleme, silme, değiştirme işlemlerini yapabilmek için bir de sorgulama dili sunmaktadırlar. Combi ve arkadaşları (2009) zamansal ve yarı yapılı veri ambarları için grafik tabanlı bir model ve gerekli bilgiyi kullanışlı bir şekilde çekmeyi sağlayan

bir sorgulama dili sunmuştur.

Bu çalışmada; veri ambarı ve OLAP uygulamalarının öznelik üzerinde olay ve işlem zamanı bilgilerini içeren, çift zamanlı bir uzantısı yapılmıştır. Veri ambarı ve zamansal veri tabanları alanlarını birbirlerine bağlayan araştırma projeleri daha önceden yapılmıştır. Ancak, şu ana kadar yapılan çalışmalarda zaman etiketleri satır üzerindedir. Bizim bilgimize göre veri ambarı ve öznelik zaman etiketi daha önce birlikte kullanılmamıştır.

Makalenin geri kalanı şu şekilde organize edilmiştir. İkinci bölüm, veri ambarları hakkında özet bilgi içerir. Üçüncü bölümde, çift zamanlı veri tabanları ve iki zamanlı etiketleme yaklaşımları ana hatları ile anlatılmıştır. Dördüncü bölümde, çift zamanlı veri ambarları için sunulan model ve uygulama detayları sunulmuştur. Beşinci bölümde, deneysel çalışma için kullanılan örnek sorgular ve test sonuçları açıklanmıştır. Altıncı bölümde çalışmadan çıkan sonuç özetlenmiş ve gelecek çalışmalarda yapılması planlanan işlere değinilmiştir.

Veri ambarları

Veri ambarı şemasında üç temel kavram kullanılır; olaylar (fact), boyutlar (dimension) ve ölçekler (measure). Bir olay, ölçeklerden ve bağlam verilerinden oluşan ilişkili veri öğelerinin bir derlemidir. Her olay tipik olarak bir iş öğesi, iş hareketi ya da iş sürecini çözümlenmekte kullanılabilecek olan bir oluşu ifade eder. Bir veri ambarında, olaylar sayısal verinin saklandığı ana tablolarda gerçekleştirilmektedir. Bir boyut, aynı türdeki görüntülere ait üye ya da birimlerin bir derlemidir. Olay tablosundaki her veri noktası, çok sayıdaki boyutların her birinden bir ve yalnızca bir üyeyle ilişkilendirilir. Boyutlar olaylar için kavramsal altyapıyı sağlarlar. Birçok çözümlenme işlemi olaylar üzerindeki boyutların etkisini sayıya dökmek üzere kullanılmaktadır. Boyutlar, üzerinde Çevrimiçi Çözümlenme İşlemi (Online Analytical Processing-

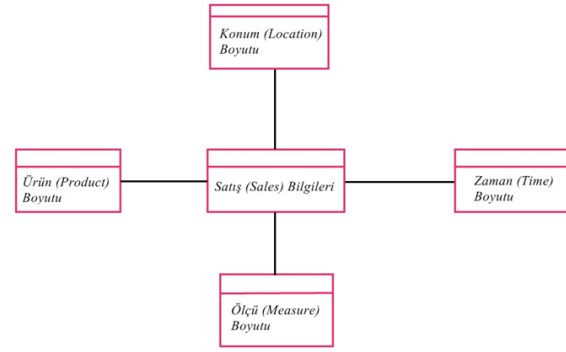
OLAP) yapılacak olan parametrelerdir. Bir boyutta birçok boyut üyesi vardır ve bunlar bir veri ögesinin yerini belirlemek üzere kullanılan ayrı bir isim ya da belirleyicidir. Örneğin, bütün şehirler, bölgeler ya da ülkeler de bir coğrafya boyutu oluştururken, bütün aylar ya da yıllar bir zaman boyutu oluşturmaktadır. Ölçek bir olayın sayısal niteliğidir. Bu nitelik, işin boyutlara göreli olarak başarımını gösterir. Sayılar değişken olarak adlandırılır. Örneğin ürün satışlarının parasal miktarı, satış hacmi, hareket miktarı gibi değerler ölçek olarak düşünülebilir. Ölçekler, boyut üyelerinin birleşimleriyle belirlenir ve olaylar üzerine yerleştirilir.

Veri ambarı şemasında kullanılacak “Yıldız modeli” ve “Kar tanesi modeli” olarak iki temel model bulunmaktadır: Yıldız modeli veri ambarı için temel yapıdır. Genellikle bir büyük merkezi tablo (olay tablosu olarak adlandırılır) ve olay tablosu etrafında yıldız biçiminde sıralanmış daha küçük tablolar söz konusudur. Şekil 1’de; merkezde bir satış bilgileri tablosu ve satış bilgileri tablosu etrafında sıralanmış zaman, ürün, konum ve ölçü boyut tablolarını gösteren, bir yıldız modeli örneği görülmektedir.

Kar tanesi modeli bazen kendi sıra düzenlerini de içeren bir ya da daha fazla boyutun ayrıştırılmasının sonucu olarak ortaya çıkar. Bir boyut tablosunun üyeleri arasındaki birçok-bir ilişkileri, bir sıra düzen oluşturacak şekilde ayrı bir boyut tablosu olarak tanımlanabilir. Örneğin, Şekil 1’deki ürün boyutu, Şekil 2’de model, kategori ve alt-kategori boyutlarına ayrıştırılmıştır. Yıldız modelinden türetilen bu model, görüldüğü gibi bir kar tanesine benzemektedir.

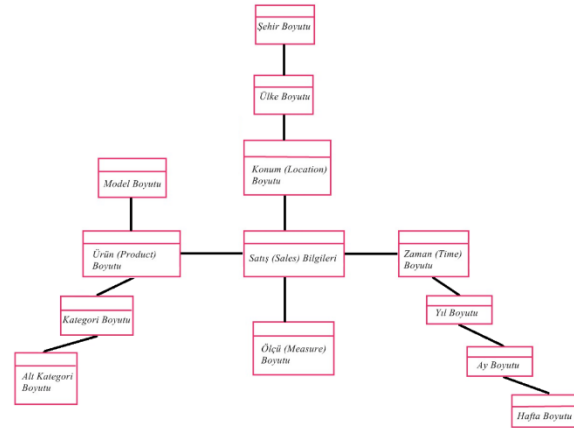
Bir boyutlu modeli görselleştirmenin en yaygın yolu bir “küp” (cube) çizmektir. Üç boyutlu bir model küp kullanılarak gösterilebilir. Boyutlu model genellikle üçten fazla boyut içerdiğinden hypercube olarak adlandırılır. Ancak bir hypercube’ün görselleştirilmesi

zor olduğundan, genel olarak kullanılan terim küptür.



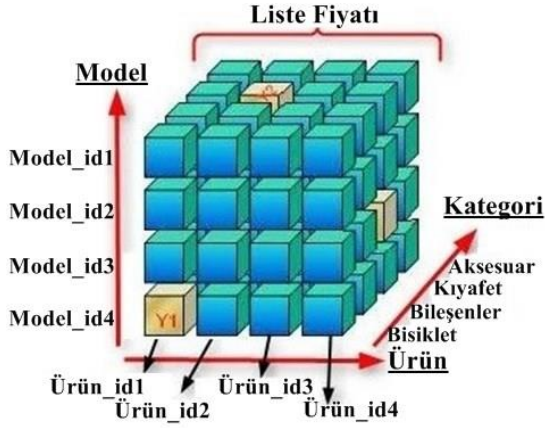
Şekil 1. Yıldız modeli.

Bu kavramları, ürünün liste fiyatını kategorisine ve modeline göre yapılan analizler ile açıklayacağız. Her bir ürünün liste fiyatı, sayı ile nitelendirilebilen bir gerçektir. Alt kategori ve model ürün boyutunun ölçüleridir. Her bir ürün tipi için farklı bir sayısal ölçü değeri vardır. Bu tip bağlantı boyutlara göre gerçeklerin organize olması ile oluşur.



Şekil 2. Kar Tanesi modeli.

Gerçek durum analizinde ortaya çıkabilecek sonuçlar boyut ve ölçü arasındaki ilişkiye ve veri ambarı şemasına bağlıdır. İhtiyaçlara uygun boyut, ölçü gibi veri ambarı nesnelere saptanır. Şekil 3 de liste fiyatının ürün, model ve alt kategori boyutlarına göre analiz edildiği çok boyutlu bir küp sunulmuştur.



Şekil 3. Çok boyutlu modelin küp görünümü sunumu.

Çift zamanlı veri tabanları

Tarihsel veri tabanları veri tabanı durumlarını nesnelerin olay zamanı ekseninde saklar, ancak nesnelerin sistemde yaşadıkları değişiklikleri içermez. İşlem veri tabanları sistem değişikliklerini saklar, ama nesneler hakkında zamansal bilgi içermez. Nesnelerin zaman içindeki değişimleri hakkında tam bilgi sahibi olabilmek için tarihsel ve işlemsel veri tabanlarının bir arada kullanılması gerekmektedir. Çift zamanlı veri tabanı bu gereksinimleri karşılar. Böylelikle, çift zamanlı veri tabanları, gerçek verilerinin değişimini zaman üzerinde doğru bir şekilde ifade etmeyi sağlar (Ben-Zvi, 1982; Golfarelli ve Rizzi, 2011).

Zamansal veri tabanlarının uygulanmasında çok bilinen iki yaklaşım vardır. Bunlardan biri satır zaman etiketlemedir ve birinci normal form ilişkilerini kullanır (1NF). Bir diğeri öznitelik zaman etiketidir ve normal form ilişkilerini içermesi gerekmez. Satır zaman etiketleme yaklaşımında zamanın satır üzerinde belirtildiği durumlarda özellikler aynı anda değişmiyorsa, her bir özellik için ayrı bir tablo oluşturmak gerekir. Zamanla değişmeyen özellikler ise başka bir ilişki ile gruplandırılır. Satırlara zaman etiketi eklemek için, dört tarihsel özellik satıra eklenir. Satır zaman etiketleme, geleneksel ilişkisel veri tabanlarının tüm avantajlarından faydalanır. Ancak zamanla değişen

her özellik için ayrı bir zaman bilgisi vardır ve ayrı bir ilişki ile gösterilmelidir. Örneğin, ürünün liste fiyatının geçerlilik aralığı, olay başlangıç zamanı (VT_LB) ve olay bitiş zamanı (VT_UB) arasında geçen süredir. Eğer bir kayıt şu an geçerli ise VT_UB değeri 'şimdi' (now) ye eşittir. Buna ek olarak, geçmiş ve şu anda gerçekleşen veri tabanı işlemlerinin ilk işlem zamanı (TT_LB) ve son işlem zamanı (TT_UB) bilgileri de tutulur. Tablo 1'de görülen MB1 ürününün eski liste fiyatı 1191.17'dir ve ilk işlem zamanı 24.09.2002'dir. MB1 ürününün liste fiyatı 01.07.2002 tarihinde değişip 1226.9 olduğunda tabloya yeni bir satır eklenmiştir. Eski liste fiyatı bilgisinin olay bitiş tarihi ve son işlem tarihi 01.07.2002 olarak değişmiştir. Yeni liste fiyatı bilgisinin ise olay başlangıç tarihi ve ilk işlem tarihi bilgileri 02.07.2002 olarak kaydedilmiştir, yeni bilgi geçerli olan bilgi olduğundan VT_LB değeri şimdiyi (now) göstermektedir.

Öznitelik zaman etiketleme durumunda normal form özellikleri kullanılmaz ve zaman etiketleri özniteliklere bağlıdır. Modellenen nesne ile ilgili tüm bilgiler tek bir satırda saklanabilir. Nesneye ait herhangi bir değer değiştiğinde, yenilenmeden etkilenmeyen diğer değerlerin (Ürün adı, ürün rengi gibi) tekrarlanması gerekmez. Zamanla değişen ve değişmeyen tüm özellikler bir bütün olarak tek bir tablo içinde modellenebilir.

Çift zamanlı atom (<işlem zamanı, olay zamanı, değer>) olarak gösterilen üçlüdür. <[VTl, VTu), [TTl, TTu), V> formundaki çift zamanlı atomda, VT_LB olay başlangıç zamanını, VT_UB olay bitiş zamanını, TT_LB işlem başlangıç zamanını, TT_UB işlem bitiş zamanını ve V nesne değeri olarak ifade edilir. Tablo 2' deki çift zamanlı atom

{<[25.09.2001, 01.07.2002], [24.09.2001, 01.07.2002], 1191.17>,

<[02.07.2002, now], [02.07.2002, now], 1226.9>}

Tablo 1. Satır Zaman Etiketli Çift Zamanlı Yaklaşım.

P. Name	Color	List Price	VT_LB	VT_UB	TT_LB	TT_UB
MB1	Black	1191.17	25.09.2001	01.07.2002	24.09.2001	01.07.2002
MB1	Black	1226.9	02.07.2002	now	02.07.2002	now
MB2	Yellow	120.43	18.10.2002	now	20.10.2002	now
MP	Red	150	30.07.2002	now	01.08.2002	now
MA	Black	35.89	18.10.2002	now	20.10.2002	now

MB1 adlı ürünün veri tabanına 24.09.2001 tarihinde yazıldığını, efektif olay başlama zamanının 25.09.2001 olduğunu belirtir. Yaklaşık olarak on ay sonra, 01.07.2002 tarihinde yapılan bir işlemle ürünün fiyatı 1226.9'a yükseltilmiştir. Bir güncelleme işlemi yapıldığında, liste fiyatı özelliğine yeni bir satır eklenir, diğer özelliklerde değişim olmaz.

Çift zamanlı veri ambarı modeli ve uygulaması

Bu çalışmada kullanılan veriler araştırma ve deney amaçlı kullanılabilen Microsoft Adventure Works (AW) veri tabanından alınmıştır. AW veri tabanında beş farklı modül bulunmaktadır. Biz bu çalışmada kendi metodumuzu Production (Ürün) modülü üzerinde uyguladık. Ürün modülü şemasını öznitelik zaman etiketli ve normal formda olmayacak şekilde iç içe tablolar halinde yeniden oluşturduk.

AW yapısına uygun olarak, bazı tablolar gereksiz tekrarları önlemek için kendileri ile ilişkili tabloların içine gömüldü. AW Ürün modülünde 25 tablo varken, oluşturduğumuz sistemde bu bilgilerin tamamını içeren 15 tablo bulunmaktadır.

Nesnel ilişkili veri tabanı bize kullanıcı tarafından yeni tip (type) tanımlama imkanı sunar. Şekil 4'de çift zamanlı atom tipinin tanımlanmasını sağlayan veri tabanı sorgusu sunulmuştur. Tip bir kere tanımlandıktan sonra, veri tabanı içinde kendiliğinden gelen tipler gibi SQL sorgularında kullanılabilir.

Kullanıcı tanımlı tipler, tablonun tamamını oluşturan bir "veri tipi" olarak tanımlanabilir. Seçme, ekleme, silme gibi veri işlemleri ilişkisel veri tabanı

tablolarındaki gibi yapılabilir. İç içe bir tablo aynı tipteki elemanların oluşturduğu sırasız bir veri koleksiyonudur. Ayrıca iç içe tablolardaki özellik sayısı herhangi bir sınırlandırma yoktur.

```
CREATE TYPE BT_NUMBER AS (
    VT_LB DATE,
    VT_UB DATE,
    TT_LB DATE,
    TT_UB DATE,
    VALUE NUMBER);
```

Şekil 4. Kullanıcı tanımlı tip olarak çift zamanlı atom BT_NUMBER tanımlama.

AW veri tabanındaki tablo yapısı analiz edilip yeniden dizayn edilmiştir. Şekil 5'de LISTPRICE, PRODUCTSELL, ve PRODUCTCOST tiplerinin sütun zaman etiketli ve yarı yapıları oluşturulması için kullanılan sorgu bulunmaktadır. Bu çift zamanlı tablolar sadece BT_NUMBER'dan oluşan tek özellikli tablolarıdır.

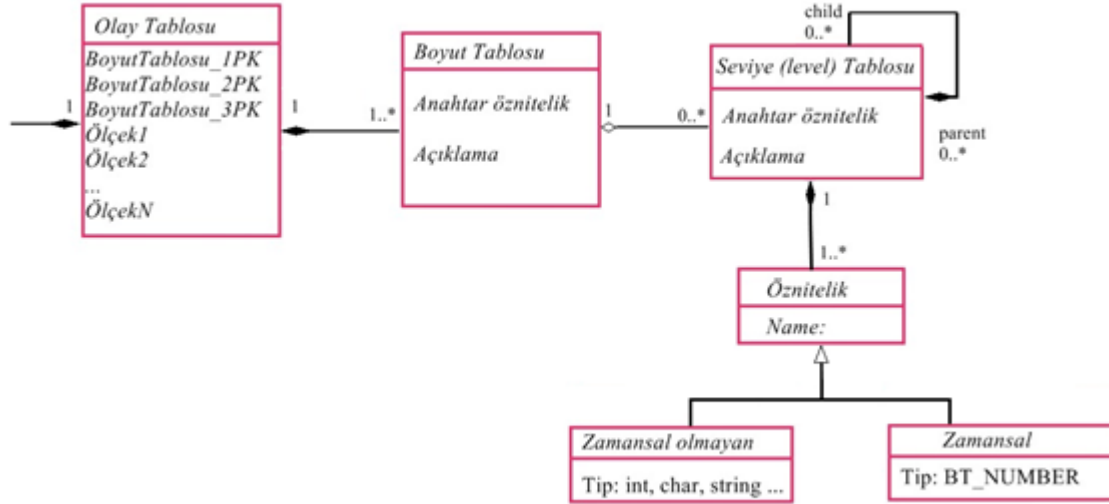
```
CREATE TYPE LISTPRICE AS TABLE OF
BT_NUMBER;
CREATE TYPE PRODUCTSELL AS TABLE OF
BT_NUMBER;
CREATE TYPE PRODUCTCOST AS TABLE OF
BT_NUMBER;
```

Şekil 5. Liste fiyatı (LISTPRICE) satırının BT_NUMBER tipinde iç içe tanımlanışı.

Kavramsal modeller, uygulamanın kullanıcı ihtiyaçlarına en yakın şekilde tanımlanmasını sağlar. Kavramsal modeller tasarımcı ve kullanıcı arasındaki iletişimin kolaylaşmasını sağlar [8].

Tablo 2. Öznitelik Zaman Etiketli Çift Zamanlı Yaklaşım.

P. Name	Color	List Price
MB1	Black	{<[25.09.2001, 01.07.2002], [24.09.2001, 01.07.2002], 1191.17 >, <[02.07.2002, now], [01.07.2002, now], 1226.9>}
MB2	Yellow	<[18.10.2002, now], [20.10.2002, now], 120.43>
MP	Red	<[30.07.2002, now], [01.08.2002, now], 150>
MA	Black	<[18.10.2002, now], [20.10.2002, now], 35.89>
TR	Black	<[30.07.2002, now], [01.08.2002, now], 22.11>

**Şekil 6.** Çok boyutlu çift zamanlı veri ambarının kavramsal modeli.

Önerilen çok boyutlu, yarı yapılı veri ambarı modeli Şekil 6' de sunulmuştur.

Bir olay tablosu üç veya daha fazla boyut tablosu ile ilişkilidir. Boyutlar bir veya birden çok seviyeden oluşur. Boyutların ve seviyelerin anahtar ve açıklayıcı alanları vardır. Özellikler adları int, real, varchar gibi veri tipleri ile ifade edilir. Seviyelerdeki özellikler için çift zamanlılık, çok özellikli olarak tanımlanan BT_NUMBER veri tipi ile sağlanır. En az bir zamansal seviyesi olan seviyeler zamansal seviyelerdir. En az bir zamansal seviye içeren boyutlar zamansal boyutlardır. Seviyeler ve boyutlar iç içe tablo olarak saklanan birçok zamanla değişen özellik barındırabilir.

AW veri tabanını analiz ettikten sonra, beş farklı küp dizayn edilmiştir.

Şekil 5'de LISTPRICE, PRODUCTSELL, ve PRODUCTCOST tiplerinin sütun zaman etiketli ve

yarı yapılı oluşturulması için kullanılan sorgu bulunmaktadır. Bu çift zamanlı tablolar sadece BT_NUMBER'dan oluşan tek özellikli tablolardır.

Kavramsal modeller, uygulamanın kullanıcı ihtiyaçlarına en yakın şekilde tanımlanmasını sağlar. Kavramsal modeller tasarımcı ve kullanıcı arasındaki iletişimin kolaylaşmasını sağlar [8]. Önerilen çok boyutlu, yarı yapılı veri ambarı modeli Şekil 6' de sunulmuştur.

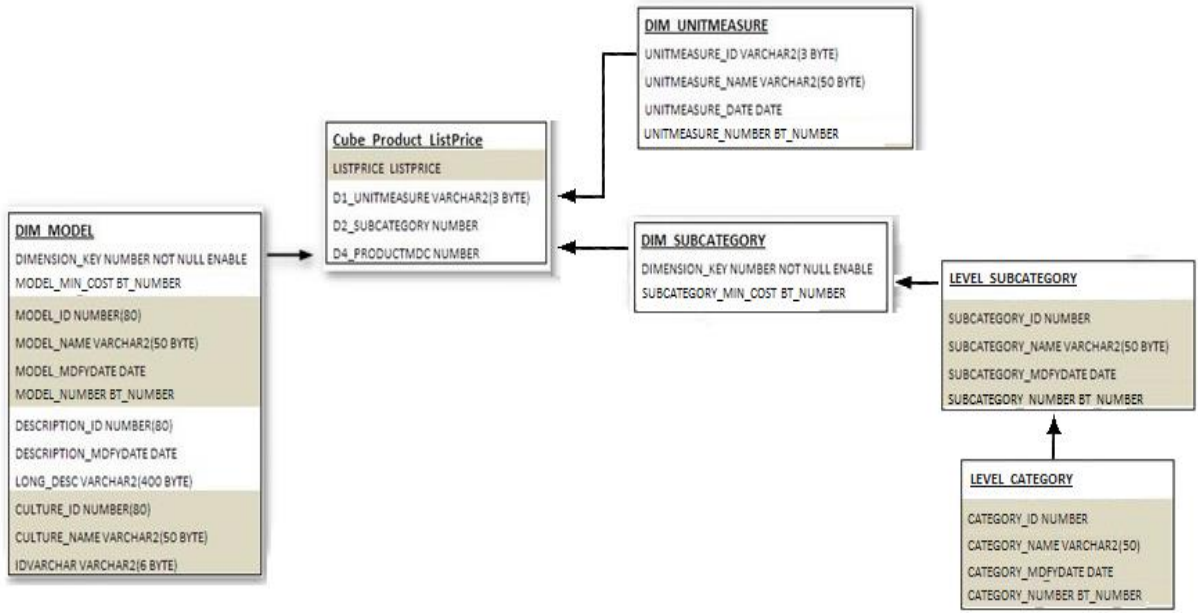
Bir olay tablosu üç veya daha fazla boyut tablosu ile ilişkilidir. Boyutlar bir veya birden çok seviyeden oluşur. Boyutların ve seviyelerin anahtar ve açıklayıcı alanları vardır. Özellikler adları int, real, varchar gibi veri tipleri ile ifade edilir. Seviyelerdeki özellikler için çift zamanlılık, çok özellikli olarak tanımlanan BT_NUMBER veri tipi ile sağlanır. En az bir zamansal seviyesi olan seviyeler zamansal seviyelerdir. En az bir zamansal seviye içeren boyutlar zamansal boyutlardır.

Seviyeler ve boyutlar iç içe tablo olarak saklanan bir çok zamanla değişen özellik barındırabilir.

AW veritabanını analiz ettikten sonra, beş farklı küp dizayn edilmiştir. Bunlar; Product, ProductListPrice, ProductCostHistory, ProductCost History2Dim, ProductListPrice2Dim ve Workorder_Order-Qty isimli küplerdir. Şekil 5 de gösterilen Cube_Product_ListPrice gerçek tablosundaki, D1_Unitmeasure, D2_Subcategory ve D3_Product MDC özellikleri boyut tabloları ile ilişki kurmaya yarayan özelliklerdir, BT_NUMBER tipindeki LISTPRICE özelliği ise ölçü değerini saklar.

Cube_Product_ListPrice küpü; Dim_Model, Dim_UnitMeasure ve Dim_Subcategory olmak üzere

üç boyuttan oluşmuştur. Dim_Model boyut tablosunda, Dimension_Key özelliği birincil anahtardır ayrıca BT_NUMBER veri tipinde Model_Min_Cost ve Model_Number adında iki özellik içerir. Dim_UnitMeasure tablosunda Unitmeasure_Id özelliği birincil anahtardır BT_NUMBER veri tipinde Unitmeasure_Number özelliği bulunur. Dim_Subcategory boyut tablosunda Dimension_Key özelliği birincil anahtardır, bunun yanında BT_NUMBER tipinde Subcategory_Min_Cost özelliği bulunur. Dim_Subcategory boyutunda iki seviye vardır. Şekil 7 de görüldüğü gibi Subcategory isimli seviyede BT_NUMBER veri tipinde Subcategory_Number adlı, Category isimli seviyede ise Category_Number adlı özellikler bulunur.



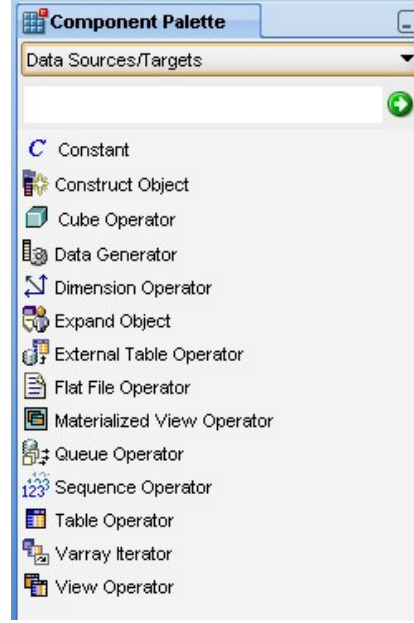
Şekil 7. Cube_Product_ListPrice tablosunu baz alan kar tanesi şeması.

Gerekli veri ambarı nesnelere Warehouse Builder aracı ile oluşturulmuştur. Veri ambarı düzenli bir şekilde oluşturulduktan sonra, sıra bu nesnelere saklanacak verileri, veri tabanından çekilen verilerle eşlemeye gelir. Warehouse Builder aracının içinde veri eşlemeyi sağlayan "Mapping" adı verilen bir araç vardır. Bu araç veri tabanındaki verileri alır ve hedef veri ambarı depolama alanına düzeni bozmayacak şekilde aktarır. Eşleme aracında veri ambarına aktarılacak verileri

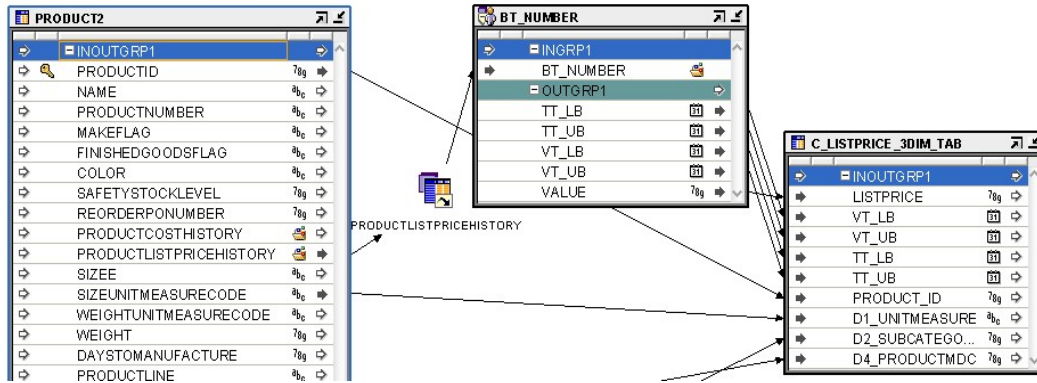
düzenlemek için, çok boyutlu nesnelere ve veri tabanı nesnelere sürüklenip bırakılabildiği bir tasarım arayüzü vardır. Verileri işlemek için gerekli her türlü bileşen, bileşen paletinde bulunur. Veri eşleme bileşen paleti Şekil 8 de gösterilmektedir.

Şekil 9 da örnek bir eşleme diyagramı gösterilmiştir. PRODUCT2 veri tabanı tablosu içindeki veriler, sağ taraftaki küp tablosuna eşlenmiştir. Veri ambarı nesnelere alanlar direkt olarak iç içe tipte

tanımlanamaz. Veri tabanında tanımlı iç içe bir tip olan BT_NUMBER tek bir alanda tutulabilirken, veri ambarında BT_NUMBER'ı oluşturan her bir özellik <VT_LB, VT_UB, TT_LB, TT_UB, VALUE> ancak ayrı birer alanda tutulabilir. Veri ambarı altyapısı henüz ne yazık ki iç içe tipleri kabullenmeye hazır değildir. Bu sebeple veritabanındaki tablolarında iç içe tipteki alanların, veri ambarı nesnelere ile eşlenirken, bazı işlemlerden geçmesi gerekir. Eşleme ara yüzünde bulunan genişleme nesnesi (Expand Object), iç içe tipleri giriş, iç içe tiplerin içindeki her bir alanı ise çıkış değeri olarak düzenler.



Şekil 8. Veri işleme bileşen paleti.



Şekil 9. Örnek bir eşleme tasarım sayfası.

Çift zamanlı veri ambarı sorguları

Sunulan modelin uygulanabilirliğini göstermek adına; aynı sonucu veren sorgular, hem ilişkisel veritabanı nesnelere kullanarak, hem de çok boyutlu çift zamanlı veri ambarı nesnelere kullanarak çalıştırılmıştır. Sorgular belirli zaman aralıklarında altışar kez çalıştırılmış ve bu altı çalıştırmanın ortalaması alınmıştır. Şekil 10 da üç farklı sorgu için, ortalama sorgulama zamanları listelenmiştir. Sorgu 3 ün diğer sorgulardan daha uzun sürece çalıştığı açık bir şekilde gözlemlenmektedir. Bunun sebebi, Sorgu 3 ün veri çektiği tablonun kayıt sayısının fazla olmasıdır.

Sorgu 1: Kategorilere, alt kategorilere ve yıllara göre ortalama bisiklet fiyatları nedir?

Aynı sonucu veren sorgular Tablo 3'te gösterildiği gibi iki farklı ortamda yazılmıştır. Veri ambarı sorgusu sonucu elde edebilmek için CUBEPRODUCTLISTPRICE küp tablosundan veri almaktadır, ayrıca her biri farklı bir sınırlandırma içinde olmak üzere iki tane iç içe sorgu içerir. İlişkisel veritabanı sorgusu aynı sonucu elde edebilmek için, dört farklı tablodan veri çeker ve bu tabloları birbiri ile ilişkilendirebilmek için üç farklı sınırlandırma koşulu içerir.

Sorgu 2: Modellere göre ortalama bisiklet fiyatları nedir?

Tablo 3. Sorgu 1, Veri ambarı ve ilişkisel veri tabanı nesnelere ile

Veri Ambarı:
SELECT VT_LB,VT_UB,TT_LB,TT_UB,ROUND(AVG(LISTPRICE),2), (SELECT CATEGORY_NAME FROM DIM_SUBCATEGORY WHERE DIMENSION_KEY= DIM_SUBCATEGORY_KEY) CATEGORY_NAME, (SELECT SUBCATEGORY_NAME FROM DIM_SUBCATEGORY WHERE DIMENSION_KEY= DIM_SUBCATEGORY_KEY) SUBCATEGORY_NAME FROM CUBEPRODUCTLISTPRICE GROUP BY VT_LB,VT_UB,TT_LB,TT_UB,DIM_SUBCATEGORY_KEY ORDER BY DIM_SUBCATEGORY_KEY;
İlişkisel Veri Tabanı:
SELECT STARTDATE AS VT_LB, ENDDATE AS VT_UL, PLP.MODIFIEDDATE AS TTLB, PLP.MODIFIEDDATE AS TT_UB, PC.NAME AS CATEGORY_NAME, PSC.NAME AS SUBCATEGORY_NAME, AVG(PLP.LISTPRICE) FROM PRODUCTION.PRODUCTLISTPRICEHISTORY PLP, PRODUCTION.PRODUCTSUBCATEGORY PSC, PRODUCTION.PRODUCT P, PRODUCTION.PRODUCTCATEGORY PC WHERE PLP.PRODUCTID=P.PRODUCTID AND PSC.PRODUCTSUBCATEGORYID=P.PRODUCTSUBCATEGORYID AND PC.PRODUCTCATEGORYID=PSC.PRODUCTCATEGORYID GROUP BY STARTDATE,ENDDATE,PLP.MODIFIEDDATE,PC.NAME,PSC.NAME;

Veri ambarı sorgusu sonucu elde edebilmek için CUBEPRODUCTLISTPRICE küp tablosundan veri almaktadır, ayrıca bir tane iç içe sorgu içerir. İlişkisel veri tabanı sorgusu aynı sonucu elde edebilmek için, üç farklı tablodan veri çeker ve bu tabloları birbiri ile ilişkilendirebilmek için iki farklı sınırlandırma koşulu

çerir. Veri ambarı sorgusu bir tablodan veri çekerken, ilişkisel veri tabanı sorgusu üç farklı tablodan verileri çekmek için daha fazla bağlantı ve arka planda daha fazla işlem yapmaktadır. Aynı sonucu veren sorgular Tablo 4'te gösterilmektedir.

Tablo 4. Sorgu 2, Veri ambarı ve ilişkisel veri tabanı nesnelere ile

Veri Ambarı:
SELECT ROUND(AVG(LISTPRICE),2), (SELECT MODEL_NAME FROM DIM_MODEL WHERE DIMENSION_KEY=DIM_PRODUCTMDC_KEY) MODEL_NAME FROM CUBEPRODUCTLISTPRICE GROUP BY DIM_PRODUCTMDC_KEY ORDER BY DIM_PRODUCTMDC_KEY;
İlişkisel Veri Tabanı:
SELECT STARTDATE AS VT_LB, ENDDATE AS VT_UL, PLP.MODIFIEDDATE AS TTLB,PLP.MODIFIEDDATE AS TT_UB, PM.NAME AS MODEL_NAME,AVG(PLP.LISTPRICE) FROM PRODUCTION.PRODUCTLISTPRICEHISTORY PLP, PRODUCTION.PRODUCTMODEL PM, PRODUCTION.PRODUCT P WHERE PLP.PRODUCTID=P.PRODUCTID AND PM.PRODUCTMODELID=P.PRODUCTMODELID GROUP BY STARTDATE,ENDDATE,PLP.MODIFIEDDATE,PM.NAME;

Sorgu 3: İş emirlerini, kategori ve alt kategoriye göre artan sıra ile listeleyiniz.

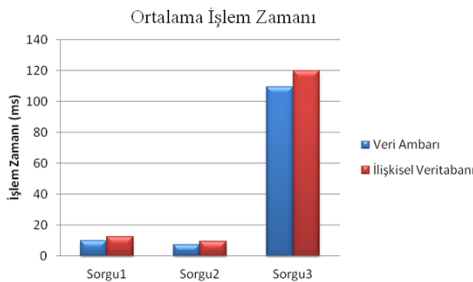
Veri ambarı sorgusu sonucu elde edebilmek için CUBE_WORKORDER_ORDERQTY küp tablosundan ve DIM_PRODUCT boyut tablosundan

veri almaktadır. Bu tabloları ilişkilendirmek için bir tane sınırlandırma koşulu içerir. İlişkisel veri tabanı sorgusu aynı sonucu elde edebilmek için, dört farklı tablodan veri çeker ve bu tabloları birbiri ile ilişkilendirebilmek için üç farklı sınırlandırma koşulu içerir. Yazılan sorgular Tablo 5'ta gösterilmektedir.

Tablo 5. Sorgu 3, Veri Ambarı ve İlişkisel Veritabanı Nesnelere ile

Veri Ambarı:
<pre> SELECT MAX(CUBE_WORKORDER_ORDERQTY.VALUE) ORDERQTY,DIM_PRODUCT.CATEGORY_NAME, DIM_PRODUCT.SUBCATEGORY_NAME, DIM_PRODUCT.PRODUCT_NAME FROM CUBE_WORKORDER_ORDERQTY, DIM_PRODUCT WHERE DIM_PRODUCT.DIMENSION_KEY= CUBE_WORKORDER_ORDERQTY.DIM_PRODUCT_KEY GROUP BY DIM_PRODUCT.CATEGORY_NAME, DIM_PRODUCT.SUBCATEGORY_NAME, DIM_PRODUCT.PRODUCT_NAME ORDER BY DIM_PRODUCT.CATEGORY_NAME , DIM_PRODUCT.SUBCATEGORY_NAME,DIM_PRODUCT.PRODUCT_NAME </pre>
İlişkisel Veri Tabanı:
<pre> SELECT MAX(WO.ORDERQTY) ORDER_QTY, PS.NAME SUBCATEGORY_NAME, PC.NAME CATEGORY_NAME, P.NAME PRODUCT_NAME FROM PRODUCTION.WORKORDER WO, PRODUCTION.PRODUCT P, PRODUCTION.PRODUCTSUBCATEGORY PS, PRODUCTION.PRODUCTCATEGORY PC WHERE P.PRODUCTID=WO.PRODUCTID AND P.PRODUCTSUBCATEGORYID=PS.PRODUCTSUBCATEGORYID AND PS.PRODUCTCATEGORYID=PC.PRODUCTCATEGORYID GROUP BY PS.NAME, PC.NAME, P.NAME ORDER BY ORDER_QTY DESC </pre>

Çok boyutlu çift zamanlı model kullanıcı tanımlı tiplerle ve iç içe bağlantılarla tasarlanmıştır; bu nedenle veri ambarında sorgular aynı işi daha az tablo kullanarak yapabilir. Diğer taraftan ilişkisel veri tabanı sorgularını işletmek için çok sayıda tablodan veri çekilir ve bu tabloları birbiri ile ilişkilendirmek için anahtar ilişkiler belirlenir. Sonuç olarak, yapılan ölçümlerde ilişkisel veri tabanı sorgularının aynı işi yapabilmek için daha çok zaman harcadıkları tespit edilmiş ve Şekil 10 da elde edilen değerler sunulmuştur. Ayrıca ilginç bir şekilde, çok boyutlu modelin oluşturduğu tablolar segmentlerde 11,392KB yer kaplarken, ilişkisel veri tabanı tabloları 34.688KB yer kaplamaktadır. Bu ikinci yaklaşımın üç kat daha fazla bellek alanı gerektirdiğini ifade eder ki bu oldukça büyük bir rakamdır.

**Şekil 10.** Sorgular için yürütme zamanları.

Sonuç

Veri ambarı; müşteri ilişkileri yönetimi sistemleri (CRM-customer relationship management), küresel e-ticaret kuruluşları, tedarik zinciri yönetimi (SCM-supply chain management), kurumsal kaynak planlama sistemleri (ERP-enterprise resource planning), kurumsal bilgi portalları, strateji yönetimi ile yakından ilişkilidir. Gün geçtikçe veri ambarı adı geçen sistemlerin ayrılmaz bir parçası haline gelmektedir.

Bu çalışma, olay ve işlem zamanlarını öznelik zaman etiketleme yöntemiyle veri ambarına uygulanmasını açıklar ve deneysel örnek modelin uygulanabileceğini gösterir. Model hem olay hem de işlem zamanlarının veri ambarlarında kullanılacağını göstermektedir. Ayrıca, değişen dünyamızın değişen bilgilerini modelleyerek veri değerlerini eylemler ile birleştirir ve eylemlerin ne zaman geçerli olduğunu belirtir. Dolayısıyla veri değerlerinin geçmişini ve değişimlerinin bütün dökümünü sağlar. Gerçek dünyayı eksiksiz ve düzgün modelleyebilmek için her iki zaman boyutu da gereklidir.

Önerilen modelin uygulama performansı hakkında fikir edinmek için farklı sorgular çalıştırdık. Sonuçlar

sunulan modelin yalın veri tabanı uygulamasından daha iyi sonuç ve daha iyi performans ortaya koyduğunu gösteriyor.

Birçok firma daha etkili ve hızlı raporlama için veri ambarı teknolojilerini kullanmaktadır. Öznitelik zaman etiketli çift zamanlı yaklaşım daha az birleştirme deyimi kullanır ve arka planda daha az işlem yapar. Bu nedenle performansı daha iyidir. Yaptığımız çalışmanın veri ambarı çalışmalarına zemin hazırlamasını umut etmekteyiz.

Kaynaklar

ABELLÓ A., MARTÍN C. The Data Warehouse: An Object-Oriented Temporal Database. JISBD, 675-684, 2003.

ATAY C. E., TANSEL A. U. Bitemporal databases: Modeling and implementation, VDM Publishing, Germany, 2009.

BEN-ZVI J. The time relational model, Doktora Tezi, University of California, Los Angeles, 1982.

CECI M., CUZZOCREA A., MALERBA D. Effectively and efficiently supporting roll-up and drill-down OLAP operations over continuous dimensions via hierarchical clustering. Journal of Intelligent Information Systems, 1–25, 2013.

COMBI C., OLIBONI B., POZZI G. Modeling and querying temporal semistructured data warehouses. New Trends in Data Warehousing and Data Analysis Annals of Information Systems, 1-25, 2009.

EDER J., KONCILIA C., MORZY T. A. model for a temporal data warehouse. OES-SEO Workshop, 48-54, 2001.

GOLFARELLI M., RIZZI S. Temporal data warehousing: Approaches and techniques, PA: Information Science Reference, 1-18, 2011.

INMON W., KRISHNAN K. Building the Unstructured Data Warehouse, Technics Publications, New Jersey, 2011.

Bundan sonraki çalışmalarda, veri ambarı farklı zaman aralıklarında çift zamanlı yapının esnekliğinden faydalanılarak analiz edilebilir. Müşterilerin satın alma alışkanlıkları zamana bağlı analiz edilebilir. Yalın verimli dönemleri saptanabilir. Bu ve benzer yararlı durumlar; veri ambarı, kullanıcı tanımlı veri tipleri ve iç içe veri saklayabilmenin faydasıyla, diğer sistemlerden daha hızlı analiz edilebilir. Sonuçlar şirketlere stratejilerini belirlemede yardımcı olacaktır.

JANET E., RAMÍREZ R., GUERRERO E. B. A model and language for bitemporal schema versioning in data warehouses, IEEE, CIC '06. 15th International Conference, 309–314, 2006.

KAUFMANN M., FISCHER P. M., MAY N., TONDER A., KOSSMANN D. Tpc-bih: A Benchmark for Bitemporal Databases. Springer International Publishing, 8391, 16-31, 2014.

MALINOWSKI E., ZIMANYI E. A conceptual solution for representing time in data warehouse dimensions. 3rd Asia-Pacific Conference on Conceptual Modeling, 45–54, 2006.

PONNIAH P. Data warehousing fundamentals for IT professionals, John Wiley & Sons, 2011.

PORTER L. M. L. Bitemporal relational databases and methods of manufacturing and use. U.S. Patent Application, 614-609, 2012.

RAVAT F., TESTE O. Supporting Data Changes in Multidimensional Data Warehouses. International Review on Computers and Software, 251-259, 2006.

SUBOTIĆ D., POŠČIĆ P., JOVANOVIĆ V. Data Warehouse Schema Evolution: State of the Art, Centra, European Conference on Information and Intelligent Systems, 18-23, 2014.

WREMBEL R. A survey on managing the evolution of data warehouses. International Journal of Database Management Systems, IJDM, 2010.