

Üç Boyutlu Nesnelerin Konumsal Veritabanında Yönetimi

Fatih DÖNER¹, Cemal BIYIK²

Özet

Konumsal Veritabanı Yönetim Sistemleri (KVTYS) büyük hacimli konumsal veri kümelerinin bir veritabanı içinde yönetimini sağlayarak aynı anda çok sayıda kullanıcının bu veriye erişimini mümkün kılmaktadır. Konumsal veri kümeleri genellikle iki boyutlu (2B) verileri içermekle birlikte günümüzde birçok uygulamada üç boyutlu (3B) veriye gereksinim duyulmaktadır. Uygulamalar için 3B veriye duyulan gereksinime paralel olarak GPS, lazer tarama, uzaktan algılama gibi tekniklerle 3B konumsal veri toplamının kolaylaşması büyük hacimli 3B verilerin etkili bir biçimde yönetimini gerektirmektedir. Bu makalede 3B konumsal nesnelerin bir Veritabanı Yönetim Sistemi (VTYS) içinde yönetimindeki aşamalar ele alınmaktadır. Bu aşamalar, 3B nesnelerin koordinatlarıyla temsili için konumsal referans sisteminin tanımlanması, veritabanında 3B nesnelerin geometrik olarak modellenmesi, 3B veriye hızlı erişim/sorgulama için 3B konumsal indeksleme, 3B konumsal sorgulamalar ve 3B verinin gösterimi konularını kapsamaktadır. Ticari amaçlı ve açık kodlu çeşitli KVTYS'leri mevcut olmakla birlikte yakın zaman önce 3B nesnelere için sunduğu destek nedeniyle Oracle Spatial konumsal veritabanı incelenmiştir.

Anahtar Sözcükler

Konumsal VTYS, 3B modelleme, 3B veri tipleri

Abstract

Management of Three Dimensional Objects in Spatial Database

Spatial Database Management Systems make it possible to manage large spatial data sets in a database that can be accessible by multiple users at the same time. Although the spatial data sets often include two dimensional data, many applications need three dimensional data. Paralleling to needs for three dimensional data, improvements in spatial data acquisition with GPS, laser scanning and remote sensing, which make it easy to collect three dimensional data, require large three dimensional data sets to be managed effectively. This paper examines the management stages of three dimensional spatial objects in a Database Management System. These stages include defining spatial reference system for representing three dimensional objects with real world coordinates, geometric modelling of three dimensional objects in database, and three dimensional spatial indexing for fast accessing/querying the three dimensional data, three dimensional spatial queries and representation of three dimensional data. Although there are several commercial and open-source spatial Database Management Systems, Oracle Spatial database is examined because of its support for three dimensional objects recently made available.

Key Words

Spatial DBMS, 3D modelling, 3D data types

1. Giriş

Konumsal Veritabanı Yönetim Sistemleri (KVTYS) büyük hacimli konumsal veri kümelerinin bir veritabanı içinde yönetimini sağlayarak aynı anda çok sayıda kullanıcının veriye erişimini mümkün kılmaktadır. Konumsal veri kümeleri genellikle iki boyutlu (2B) verileri içermekle birlikte artık günümüzde kent planlama (ZLATANOVA 2000; KOLBE vd. 2006) afet yönetimi (LAN vd. 2007; LEE ve ZLATANOVA 2008), kadastro (DÖNER ve BIYIK 2007; LEMMEN ve van OOSTEROM 2003; DONER vd. 2008) hava/gürültü/koku kirliliği modelleme (STOTER vd. 2008), altyapı tesisleri (DU vd. 2006; BITENC vd. 2008; XU vd. 2008; KARATAŞ 2007) vb. birçok uygulamada üç boyutlu (3B) veriye gereksinim duyulmaktadır. Arazi kullanımının giderek yoğunlaşması yanında GPS, lazer tarama, uzaktan algılama gibi tekniklerle 3B konumsal veri elde etmenin kolaylaşması sonucunda 3B konumsal veriye olan talep ve 3B uygulamalara olan ilgi giderek artmaktadır (APEL 2006).

Son yıllara kadar üçüncü boyutla ilgili araştırmalar CAD sistemleri, sanal gerçeklik, bilgisayar oyunları gibi alanlarla sınırlı olmuştur (SCHNEIDER ve EBERLY 2002). Konumsal veri toplamadaki ilerlemelere paralel olarak özellikle arazi ve kent yönetimi için 3B veriye duyulan gereksinim büyük hacimli 3B verilerin etkili bir biçimde yönetimini gerektirmektedir. Bununla birlikte, yakın bir zamana kadar dünyayı modelleyecek 3B veri tipleri ticari amaçlı konumsal VTYS'leri tarafından desteklenmemekteydi. Çözüm yolu olarak, 3B konumsal nesnelere modellemek için VTYS içinde 2B poligon gibi elemanların kullanımı tercih edilmiştir. Bu yaklaşımın belli başlı olumsuzlukları 3B konumsal nesnelerin gerçek anlamda veritabanında depolanamaması, veritabanındaki nesnelere temsil eden geometrik yapıların geçerliliğinin kontrol edilememesi, 3B nesnelere arasındaki ilişkilerin sorgulanamaması ve aynı koordinatların birçok kez kayıt edilmesi şeklinde sıralanabilir (ARENS vd. 2005).

3B veri tipi ve 3B konumsal sorgulamaların ticari amaçlı bir VTYS tarafından desteklenmesi ancak 2007 yılından sonra gerçekleşebilmiştir. Bu tarihten önce konumsal veriler 3B koordinatları olan nokta, çizgi, poligonlar kullanılarak modellenmekteydi. Bazı istisnalar dışında (PostGIS yazılımının 3B konumsal operatörleri gibi) konumsal sorgulamalar 2B ile sınırlı olarak gerçekleştirilebilmiştir. Yukarıda söz edilen

¹ Arş. Gör., Gümüşhane Üniversitesi, Müh. Fak., Jeodezi ve Fotogrametri Mühendisliği Bölümü, 29000, Gümüşhane.

² Prof. Dr., KTÜ, Mühendislik Fak., Jeodezi ve Fotogrametri Mühendisliği Bölümü, 61080, Trabzon

olumsuzlukları ortadan kaldırmak ve artan ihtiyaca cevap vermek amacıyla Oracle, Oracle Spatial (OS) ürünüde kullandığı, SDO_GEOMETRY isimli veri modelini 3B verileri de modelleyecek şekilde geliştirmiştir (ORACLE 2007).

Bu makalede, 3B nesnelerin konumsal bir veritabanında (OS) modellenmesindeki aşamalar incelenmektedir. Bölüm 2’de 2B ve 3B nesnelerin konumsal referans sisteminde tanımlanması ele alınmaktadır. Bölüm 3’de 3B nesnelerin veritabanında depolanmasında kullanılan geometri modeli açıklanmaktadır. Bölüm 4 ve Bölüm 5’te sırasıyla konumsal veritabanında tutulan 3B nesnelere için indeksleme ve sorgulama modelleri incelenmektedir. Bölüm 6’da veritabanındaki 3B nesnelere erişim ve gösterim için mevcut olanaklar değerlendirilmiştir. Makale sonuç bölümüyle son bulmaktadır.

2. Konumsal Referans Sisteminin Tanımlanması

3B nesnelerin veritabanında yönetimindeki ilk aşama Konumsal Referans Sistemi (KRS) tanımlamadır. KRS, konumsal veritabanında depolanan koordinat kümeleri arasındaki ilişkiyi belirleyerek nesnelere konumlarıyla temsil etme imkânı sağlamaktadır. Veritabanında depolanan her konumsal veri bir KRS ile ilişkili olmak zorundadır. Bu KRS, coğrafi referanslı olabileceği gibi bir yerel kartezyen koordinat sistemi de olabilir. Dik koordinat sistemi olarak da isimlendirilen kartezyen koordinat sisteminde bir noktanın koordinatları, orijinden itibaren 2B veya 3B olarak eksenler boyunca uzaklıkların ölçülmesiyle belirlenir. Coğrafi koordinatlar ise veritabanında, genelde elipsoidal enlem ve boylam değerleriyle açısız olarak ifade edilip belirli bir jeodezik datuma göre tanımlanırlar. Konumsal veritabanında coğrafi koordinatlarla çalışmak tercih edildiğinde bir jeodezik datumun kullanılması zorunludur.

OS veritabanı 3B KRS’ni de desteklemektedir. OS’da 3B jeodezik KRS için iki kategori bulunmaktadır. Bunlar elipsoit yüksekliğini temel alan ve yer gravitesi bazlı yüksekliği temel alan KRS’dir. Elipsoit yüksekliğini temel alan KRS, enlem ve boylam değerleriyle birlikte referans elipsoidinde belirlenen yükseklikleri içermektedir. Yer gravitesi bazlı yüksekliği temel alan KRS, coğrafi koordinatlar ya da 2B projeksiyon koordinatlarıyla birlikte yer gravitesi bazlı yükseklikleri tanımlamaktadır. Elipsoit yüksekliklerini temel alan bir KRS kullanılması konumsal veritabanında gerçekleştirilen hesaplamalarda matematiksel tutarlılığı ve etkinliği önemli ölçüde artırmaktadır (KOTHURI vd. 2007).

Bunun yanında, 3B koordinat dönüşümleri de OS konumsal veritabanında desteklemektedir. 3B dönüşümler iki şekilde gerçekleştirilir. Birincisi 3B datum dönüşümleri ikincisi ise yer gravitesi bazlı ve elipsoit yükseklikleri arasındaki dönüşümdür. Dönüşüm işlemlerinde üçüncü boyut ihmal edilmediğinden 2B dönüşümlerle karşılaştırıldığında doğruluk artacaktır. Bununla birlikte, özellikle yer gravitesi bazlı yüksekliklerle elipsoit yükseklikleri arasındaki dönüşüm işlemlerinin gerçekleştirilmesi yüksek dereceli polinomlar ve doğrusal olmayan enterpolasyonlar gerektirdiğinden oldukça karmaşık ve zaman alıcıdır (ORACLE 2007).

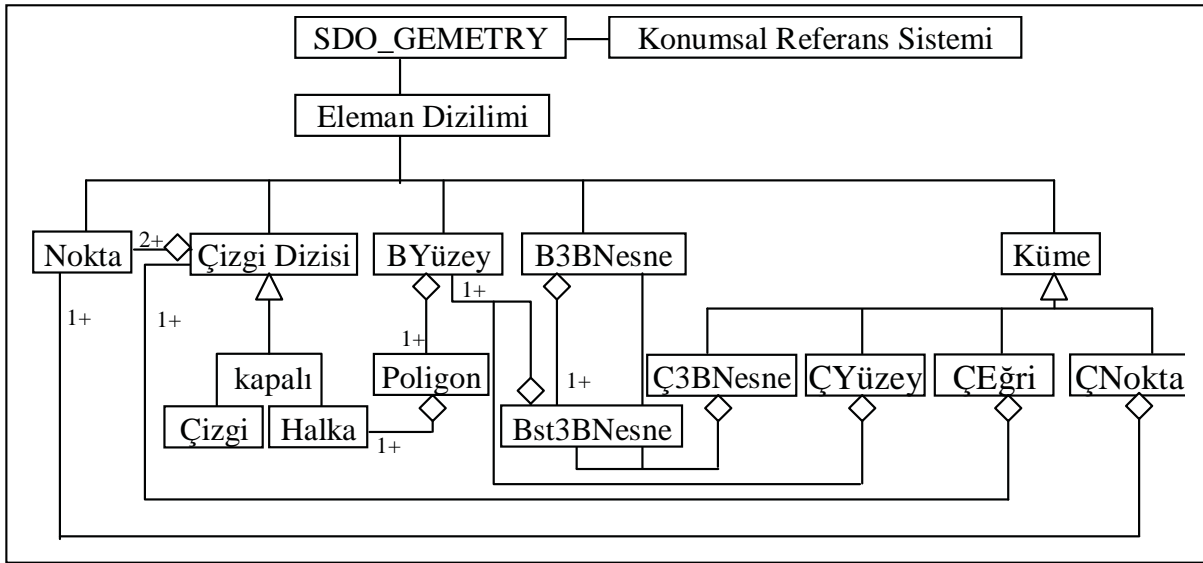
3. 3B Geometri modeli

OS’da 3B verinin depolanması için kullanılan SDO_GEOMETRY veri modeli Şekil 1’de gösterilmiştir. SDO_GEOMETRY, bir ya da çok sayıda elemanın diziliminden oluşmaktadır. Bu eleman dizilimi bir noktayı, çizgi dizisini, yüzeyi, 3B bir nesneyi ya da bunların birleşimini temsil etmek için kullanılabilir. İki ya da daha fazla sayıda nokta bir çizgi dizisini oluşturur. Kapalı bir çizgi dizisi halka olarak adlandırılır. Aynı düzlemde bulunan bir ya da daha fazla sayıda halka poligonu oluşturmaktadır. Bir ya da daha fazla sayıda poligon yüzey yapısını temsil eder. Eğer yüzey birden fazla poligondan oluşuyorsa ve paylaşılan kenarlarla bağlantılı ise birleşik yüzey adını alır. Bir birleşik yüzeyin iki veya daha fazla dış halkası aynı veya farklı düzlemde olabilir. Bir dış yüzey ve bir ya da birden çok iç yüzey, eğer tüm bu yüzeylerin birleşimi kapalı bir hacmi kuşatıyorsa, basit 3B nesne oluşturur. Bir ya da daha fazla birleşik basit 3B nesne, birleşik 3B nesneyi oluşturur. Herhangi bir birleşik 3B nesne, paylaşılan poligonlar kaldırılmak suretiyle basit 3B nesne olarak temsil edilebilir. Bir ya da daha fazla sayıda nokta topluluğu çoklu noktaları, bir ya da daha fazla sayıda 3B nesne topluluğu çoklu 3B nesneyi, bir ya da daha fazla yüzey topluluğu çoklu yüzeyi oluşturur. Bu çoklu elemanlar birbirlerinden ayrılabilecekleri gibi paylaşılan sınırlarıyla birbirlerine temas halinde de olabilirler (KOTHURI vd. 2007; KAZAR vd. 2008).

Başta bina olmak üzere kesin sınırlara sahip nesnelere veritabanında basit 3B nesne, birleşik 3B nesne ve çoklu 3B nesne olarak temsil edilebilirler. Bunun yanında, yüzeyleri de konumsal veritabanında temsil etmek ve yönetmek mümkündür. Bir yüzey hacim değil bir alan tanımlar ve 2B ya da 3B olabilir. Yüzey modellerinin en yaygın kullanımlarından biri arazinin yükseklik modelinin elde edilmesidir. Yükseklik yüzeyleri, arazinin temsil edilmesi yanında düşey boyuttaki nesnelere arazi yüzeyi arasındaki ilişkiyi belirlemek için de gereklidir (TSE vd. 2004).

OS konumsal veritabanında yüzeyler SDO_GEOMETRY veya çok büyük veri kümeleri için SDO_TIN veri modeli kullanılarak modellenirler. SDO_TIN modelinde yüzey bir üçgenler ağı şeklinde modellenir. Bu şekilde üçgen ağlarıyla modellenen yüzeyler TIN (düzensiz üçgenler ağı= Triangulated Irregular Networks) olarak adlandırılır. TIN modeli, yüzeyi sürekli ve üst üste çakışmayan üçgenler olarak temsil eder. Her bir üçgende yüzey bir düzlem ile temsil edilir. Bu üçgenler yüzey modelini temsil etmek için seçilen noktalardan oluşturulur (KAZAR vd. 2008).

Oracle veritabanındaki SDO_GEOMETRY nesnesi SDO_GTYPE, SDO_ELEM_INFO ve SDO_ORDINATES gibi özniteliklere sahiptir. SDO_GTYPE geometri tipini tanımlar. SDO_ORDINATES geometriyi oluşturan kırıkların koordinatlarını saklar. SDO_ELEM_INFO elemanların dizilimini belirtir. Her bir dizilim SDO_ORDINATES alanında depolanan koordinatların birbirleriyle nasıl ilişkili olduğunu belirtir. Şekil 2’deki örnek, iki tane dikdörtgen poligondan oluşan birleşik bir yüzey geometrisinin OS’de oluşturulmasını tanımlayan SQL kodudur. Diğer 3B veri tiplerinin gösterimi Şekil 3’de verilmiştir (KOTHURI vd. 2007).



Şekil 1. OS SDO_GEOMETRY veri modelinin sınıf diyagramı (Not: B kısaltması Birleşik, Bst kısaltması Basit, Ç kısaltması Çoklu anlamında kullanılmıştır)

```

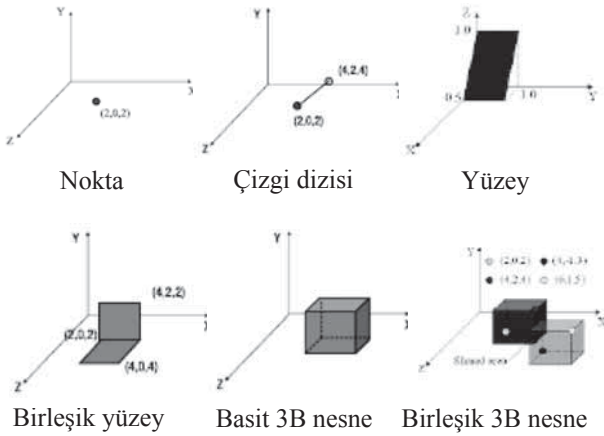
SDO_GEOMETRY(
    3003,          -- SDO_GTYPE: 3B yüzey
    NULL, NULL    -- SDO_SRID: koordinat sistemini tanımlayan ID değeri

SDO_ELEM_INFO_ARRAY(
    1,          --ilk elemanın koordinat diziliminde başlangıç sırası
    1006,      --eleman tipi, birleşik yüzey
    2,          --birleşik yüzeyi oluşturan poligon elemanının sayısı
    1,          --birleşik yüzeyin ilk poligonunun koordinat dizilimindeki başlan-gıç sırası
    1003,      --poligon elemanının tipi
    3,          --dikdörtgenin iki köşe koordinatları kullanılarak tanımlanacağını ifade eder
    7,          --birleşik yüzeyin ikinci poligonunun koordinat dizilimindeki baş-langıç sırası
    1003,      --poligon elemanının tipi
    3,          --dikdörtgenin iki köşe koordinatları kullanılarak tanımlanacağını ifade eder

SDO_ORDINATE_ARRAY ( --kırıkların gerçek koordinatlarını saklar)
    2,0,2,     --birinci poligonu tanımlayan ilk köşe koordinatı
    4,2,2,     --birinci poligonu tanımlayan ikinci köşe koordinatı
    2,0,2,     --ikinci poligonu tanımlayan ilk köşe koordinatı
    4,0,4,     --ikinci poligonu tanımlayan ikinci köşe koordinat

```

Şekil 2. OS'da yüzey geometrisini tanımlayan SQL kodu



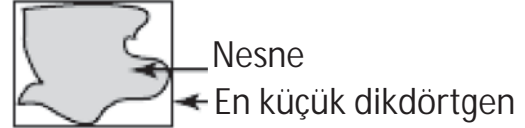
Şekil 3. OS veritabanında desteklenen 3B veri tipleri

4. 3B nesnelere için Konumsal İndeks Modeli

3B veri yönetiminin önemli konularından biri de konumsal indekslemedir. Konumsal indeksler VTYS içerisinde özellikle konumsal operatörler kullanıldığında sorgulamaları hızlandırmak için gereklidir. İndeks kullanmaksızın gerçekleştirilecek sorgulamalarda, konumsal veritabanındaki her bir kayıdı, sıralı olarak taranması gerekecektir. İndeks kullanımı veriyi bir sorgu ağacı içinde düzenleyerek belirli bir veriye hızlı bir şekilde ulaşmayı sağlamaktadır. Farklı indeks tipleri çeşitli VTYS'ler içerisinde kullanılmaktadır. Bunlardan en önemlileri B-Tree, R-Tree ve GiST (Generalized Search Trees) indeksleridir.

B-Tree indeksler tek bir eksen boyunca sıralanabilecek numaralar, harfler, tarihler gibi veriler için kullanılmaktadır. Konumsal veri böyle bir sıralama için uygun değildir. Şöyle ki; (0,0), (0,1), (1,0) koordinat çiftlerinden hangisinin daha büyük olduğunu bilmek olanaksızdır. Bu nedenle, B-Tree indeks kullanımı konumsal veriler için tercih edilmemektedir. GiST indeks modeli veriyi belirli bir bölge içinde olanlar, üst üste binenler gibi bölümlere ayırarak işler. GiST indeks modeli de konumsal VTYS'lerde kullanılmaktadır. Önemli üstünlüklerinden biri indeks sütunlarında boş değerlerin bulunmasına izin vermesidir. Bir diğer üstünlüğü ise diğer indeks tiplerine göre çok daha büyük veri kümelerini yönetebilmesidir. Eksikliği, büyük miktarda veriler için indeks oluşturmaya çok zaman alması ve 3B veriler için bu indeksin mevcut olmayışıdır. R-Tree indeksler veriyi dörtgenlere dörtgenleri alt dörtgenlere, alt dörtgenleri tekrar alt dörtgenlere bölerek çalışırlar. Birçok konumsal veritabanında R-Tree indeksler kullanılmaktadır. Bu indeksleme modelinin 3B veri için kullanımını OS'ın 11g sürümünde mümkündür (ORACLE 2007).

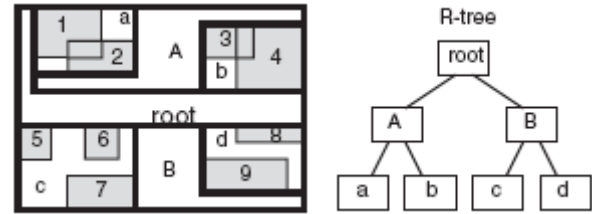
Konumsal R-Tree indeks konumsal veriyi dört boyuta kadar indeksleyebilen bir indeks türüdür. R-Tree indeks her bir konumsal nesne için bu nesneyi kuşatan en küçük dikdörtgeni belirleyerek çalışır (Şekil 4).



Şekil 4. Bir geometriyi çevreleyen en küçük dikdörtgen

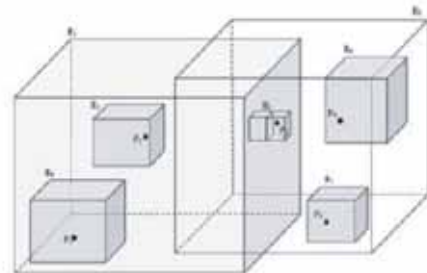
Çok sayıda nesne içeren katmansal verilerde R-Tree indeksi bu konumsal nesnelere çevreleyen en küçük dikdörtgenler topluluğundan oluşur. Şekil 5'te R-Tree indeks modelinin hiyerarşik yapısı gösterilmektedir. Buna göre (ORACLE 2007):

- 1'den 9'a kadar numaralandırılmış nesnelere aynı katman içindedir.
- a, b, c ve d; R-Tree indeks yapraklarını oluşturur. Bu yapraklar nesnelere kuşatan en küçük dikdörtgenlerden oluşurlar. Örneğin a yaprağı, 1 ve 2 numaralı nesnelere kuşatan en küçük dikdörtgeni, b yaprağı ise 3 ve 4 numaralı nesnelere kuşatan en küçük dikdörtgeni içerir.
- A ile temsil edilen bölüm a ve b yapraklarının belirlediği nesneyi çevreleyen en küçük dikdörtgeni içerir. Benzer şekilde B, c ve d yapraklarının en küçük dikdörtgeninden oluşur.
- Kök (root) ise A ve B bölümlerinin, yani tüm veri alanını kuşatan en küçük dikdörtgeninden oluşmaktadır.



Şekil 5. R-Tree indeksin hiyerarşik yapısı

R-Tree indeks modeli, verinin 2, 3 ve 4 boyutu için oluşturulabilir. İndeksin oluşturulması sırasında bir tanımlama yapılmamışsa indeks 2B olarak kabul edilir. Üçüncü boyutu destekleyen konumsal operatörlerin çalışması için indeksin 3B olması gereklidir. İki'den fazla boyutlu verilerde *sdo_indx_dims* parametresi kullanılarak konumsal indeksin boyutu belirlenir. Şekil 6, OS veritabanında 3B konumsal veriler için R-Tree indeks modelinin hiyerarşik yapısını göstermektedir (KHUAN vd. 2008).

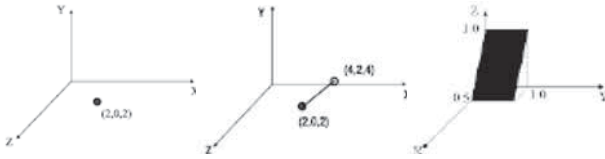


Şekil 6. R-Tree konumsal indeks modelinin 3B gösterimi

5. Sorgulama Modeli

OS, konumsal sorgulamaların sonuçlarını bulmada iki-parçalı bir sorgulama modeli üzerinde çalışır. Her aşamada sorgulama sonucuna ulaşmak için farklı işlemler gerçekleştirilir. Bu iki aşamanın birleşimi kesin sonucu vermektedir. Birinci süzgeçleme ve ikinci süzgeçleme olarak adlandırılan bu iki aşama (Şekil 7) şu şekilde çalışır (ORACLE 2007):

- Birinci süzgeçleme, büyük bir veri kümesinden sorgu sonucunun olası kayıtlarının seçilerek hızlı bir şekilde ikinci süzgeçleme aşamasına geçirilmesini sağlar. Birinci süzgeçleme, nesnelerin yaklaşık değerleriyle karşılaştırılmasını yaparak işlem karmaşıklığını en aza indirir. Yaklaşık değerler kullanıldığından birinci süzgeçleme ile elde edilen nesnelere kesin sonuç olmayıp her zaman kesin sonuçtan daha fazla sayıda nesne içerirler.
- İkinci süzgeçleme, birinci süzgeçleme sonucunda elde edilen nesnelere hesaplamalar yapar. İkinci süzgeçleme konumsal sorgunun kesin sonucunu döndürür. İkinci aşamadaki hesaplama işlemleri daha karmaşık olmakla birlikte, tüm veri kümesi yerine sadece birinci süzgeçleme sonucunda elde edilen verilere uygulanırlar.



Şekil 7. Konumsal veritabanının sorgulama modelinde birinci ve ikinci süzgeçleme arasındaki ilişki

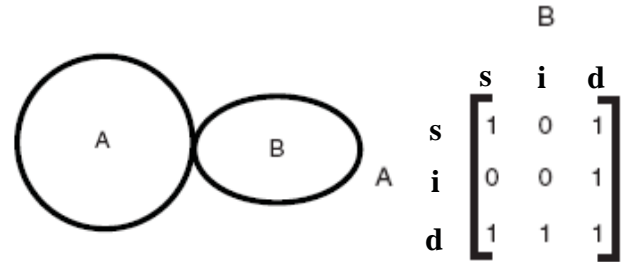
Konumsal veritabanında sorgulamaların gerçekleştirilmesinde konumsal operatörler veya konumsal fonksiyonlar (geometri işleme fonksiyonları) kullanılabilir. Konumsal operatörler, geometri sütunu üzerine tanımlanmış konumsal indeksleri kullanarak çalıştıklarından veritabanında en iyi performansı sağlarlar. Konumsal operatörlerin ilk parametresi araştırılacak konumsal veri sütununu belirlerken ikinci parametre sorgulama penceresini tanımlar. Konumsal fonksiyonlar ise konumsal indeks gerektirmez, eğer indeks oluşturulmuşsa da dikkate alınmazlar. Bu nedenle sorgulama performansı için gereksinim duyulmadıkça konumsal fonksiyonlar yerine konumsal operatörler kullanılmalıdır (ORACLE 2007).

OS'da SDO_ANYINTERACT, SDO_RELATE, SDO_FILTER, SDO_NN ve SDO_WITHIN_DISTANCE konumsal operatörleri gerçekleştirdikleri işlemlerde üçüncü boyutu da dikkate alırlar. SDO_ANYINTERACT, veritabanı tablosundaki herhangi bir nesnenin belirli bir nesneyle topolojik ilişkisinin olup olmadığını kontrol eder. SDO_RELATE, belirli bir bölge içindeki nesnelere için veya bir nesne çifti için aralarındaki etkileşimi belirler. SDO_FILTER, belirli bir bölgedeki nesnelere veya nesne çiftleri için etkileşim içinde olan olası nesnelere seçer. SDO_NN, en yakın komşu nesneyi belirler. SDO_WITHIN_DISTANCE, bir nesneye belirli bir

uzaklık içerisinde bulunan konumsal nesnelere belirler. Bunun yanında, SDO_GEOM.SDO_AREA (bir nesnenin alanını belirler), SDO_GEOM.SDO_VOLUME (bir nesnenin hacmini hesaplar), SDO_GEOM.SDO_DISTANCE (nesnelere arası uzaklığı belirler), SDO_GEOM.SDO_LENGTH (bir nesnenin uzunluğunu veya çevresini belirler) gibi konumsal fonksiyonlar da 3B veri için destek sunmaktadırlar.

Konumsal sorgulamalar ile aslında konumsal nesnelere temsil eden geometrik yapılar arasındaki ilişkiler belirlenmektedir. En yaygın konumsal ilişkiler topoloji ve uzaklığı temel alanlardır.

Veritabanında nesnelere arasındaki topolojik ilişkileri sınıflandırmak için 3x3'lük bir matris modeli kullanılır. Bu modelde, her bir nesne iç, sınır ve dış olmak üzere üç bölüme içerir. Sınır, iç ve dış bölümleri birbirinden ayıran nokta veya çizgilerden oluşur. A ile temsil edilen bir nesnenin sınırı s, içi i, dışı da d ile temsil edilecek olursa, bir nesne çiftinin bu üç bölümleri arasında dokuz etkileşim söz konusu olur. İkili bölümler arasındaki ilişkiler boş (0) veya boş olmayan (1) kesimler içerirler. Şekil 8'de birbirine komşu iki poligon için sınır, iç ve dış bölümlerinin birbirleriyle olan ilişkilerini tanımlayan 3x3'lük matris verilmektedir. Konumsal veritabanında 3B nesnelere arasındaki topolojik ilişkiler Şekil 9'da gösterilmektedir. OS'da 3B nesnelere arasındaki topolojik ilişkileri belirlemek için SDO_RELATE konumsal operatörü kullanılmaktadır (KOTHURI vd. 2007).



Şekil 8. İki nesne arasındaki ilişkiyi belirlemede kullanılan matris

Şekil 9. 3B nesnelere topolojik ilişkilerinin gösterimi

Uzaklığa göre konumsal nesnelere arasındaki ilişkinin belirlenmesinde iki nesnenin birbirinden belirli bir uzaklıkta olup olmadıkları incelenir. A ve B nesnelere arasındaki uzaklık ilişkisi belirlenecekse, öncelikle referans nesne olan B etrafında belirlenen uzaklık değeri dikkate alınarak bir tampon bölge oluşturulur. Daha sonra A nesnesi ile oluşturulan tam-

pon bölgenin ayrık olmadığı kontrol edilir. Şekil 10'da nokta, çizgi ve poligon veri tipleri için uzaklık değeri ile oluşturulan bölgeler gösterilmektedir.



Şekil 10. Nokta çizgi ve poligon için oluşturulan bölgeler.

Şekil 10'daki kesikli çizgiler bölgeleri göstermektedir. Şekilde sağda bulunan poligonun içinde bir boşluk bulunmaktadır. Büyük dikdörtgenin dışındaki kesikli çizgi dış sınıra ait bölgeyi temsil etmektedir. Küçük dikdörtgenin içindeki kesikli çizgi ise iç sınıra ait bölgeyi gösterir. Yukarıda sözü edilen uzaklık ilişkisi kullanılarak referans nesneden belirli bir uzaklık şartını sağlayan tüm nesnelere elde edilebilir. OS veritabanında SDO_WITHIN_DISTANCE konumsal operatörü uzaklığa göre konumsal ilişkilerin belirlenmesinde kullanılır. Belirli sayıda nesne ile konumsal sorgulamanın kısıtlanması isteniyorsa SDO_NN operatörü kullanılır.

6. 3B Nesnelere Erişim ve Gösterim

Gösterim olmaksızın veritabanında gerçekleştirilen sorgulamalar sadece sayı ve karakterlerden oluşacak ve kullanıcılar açısından temsil edilen bu bilgiyi anlamak oldukça zor olacaktır. VTYS'leri sadece veri kümelerinin yönetimi için araçlar sunduklarından, bu verinin gerçeğe yakın bir gösteriminin yapılabilmesi için farklı uygulamalara gereksinim vardır. Bu nedenle, VTYS'deki konumsal verinin bir gösterim aracı ile ilişkilendirilerek grafik olarak görüntülenmesi gereklidir. 3B konumsal veri VTYS içinde konumsal veri sütununda tutulmaktadır. Gösterimi gerçekleştirecek uygulamanın bu sütuna erişerek 3B veriyi gösterim için elde edebilmesi bir bağlantının mevcudiyetini gerektirir. OS'de depolanan 3B konumsal nesnelerin gösterimi için iki farklı yöntemden yararlanılabilir (STOTER ve ZLATANOVA 2003):

- Üçüncü boyutun gösterimini destekleyen CBS veya CAD uygulamalarından yararlanarak veritabanı ile bir bağlantı gerçekleştirmek.
- VRML (Virtual Reality Modelling Language) kullanılarak veritabanındaki 3B veri tipleri ile VRML söz dizimi arasında dönüşüm yapmak.

Her iki yöntemin de bazı üstünlükleri ve eksiklikleri bulunmaktadır. Örneğin, VRML'nin kullanımı ek işlem adımlarının tamamlanmasını gerektirir. 3B nesnelere VRML dosyalarına dönüştürülmelidir. Bu işlem için ilk olarak, VRML dosyası SQL dosyası olarak kayıt edilir. Daha sonra, bu SQL dosyası veritabanı araçları (Oracle SQL-loader gibi) kullanılarak bir tablo oluşturmak için VTYS ortamına yüklenir. 3B nesne geometrileri bu tabloya eklenir ve VRML dosyasına Internet üzerinden erişilebilir. VRML dosyasına web araçlarıyla erişimin mümkün olması sayesinde gösterim ve veri

dönüşümü/transferi kolaydır. Bununla birlikte, bu yaklaşımda VRML dosyası veritabanı ortamının bir parçası değildir. Ayrıca, veri hacmi büyük olduğunda nesne geometrilerinin VRML dosyalarına dönüşümü sırasında 3B gösterimin performansı düşer. Bu yaklaşımda karşılaşılan eksiklik CBS/CAD uygulamalarının sağladığı gösterim araçlarının doğrudan VTYS ile entegre edilmesiyle giderilebilir. Birçok CBS/CAD (ArcScene, Microstation gibi) uygulaması VTYS ortamında depolanan 3B veriye erişerek 3B gösterim olanağı sunmaktadır. Bununla birlikte, OS veritabanındaki hacimsel nesnelerin gösterimi CBS/CAD yazılımları tarafından henüz desteklenmemektedir. Bu nedenle, hacimsel bir nesnenin, gösterim için çok sayıda 3B nesneye dönüştürülmesi gerekli olmaktadır. Örneğin küp şeklinde temsil edilen bir nesneyi görüntüleyebilmek için küpün, altı adet poligon kullanılarak temsili gereklidir (ARENS vd. 2005).

3B gösterimde bir diğer önemli nokta da verinin anlaşılabilirliğidir. Özellikle CBS araçlarının sunduğu olanaklarla yüksek kaliteli gösterimler elde etmek mümkündür. Nesnelerin geometrik temsilleri yerine bir çeşit resimlerini kullanma veya farklı ayrıntı düzeyleri gibi yöntemlerden faydalanarak 3B verinin gösteriminde kaliteyi artırmak mümkündür. Bu gibi tekniklerin amacı, nesnelere yakınlıkla daha fazla ayrıntı, uzakta daha az ayrıntı göstermektedir. Bunların yanında, 3B gösterimin bilgisayar ekranında ve kağıt ortamında daha gerçekçi olabilmesi için renklendirme, aydınlatma/karartma, gölgelendirme, doku kullanımı gibi tekniklerden yararlanılabilmektedir (ZLATANOVA vd. 2002).

7. Sonuç

3B konumsal veri toplamanın kolaylaşmasına paralel olarak bu verileri kullanan uygulamaların da yaygınlaşması, 3B konumsal nesnelerin etkili bir şekilde yönetimi gereksinimini gündeme getirmiştir. Bu makalede 3B konumsal nesnelerin bir VTYS'de yönetimindeki aşamalar ele alınmıştır. Bu aşamalar, 3B nesnelerin koordinatlarıyla temsili için KRS'nin tanımlanması, veritabanında 3B nesnelerin geometrik olarak modellenmesi, 3B veriye hızlı erişim/sorgulama için 3B konumsal indeksleme, 3B konumsal sorgulamalar ve 3B verinin gösterimi konularını kapsamaktadır. Ticari amaçlı ve açık kodlu çeşitli konumsal VTYS'leri mevcut olmakla birlikte yakın zaman önce 3B nesnelere için sunduğu destek nedeniyle Oracle Spatial konumsal veritabanı incelenmiştir. Önemli ilerlemeler sağlanmasına rağmen, 3B nesnelerin yönetiminde gelecekte ele alınması gereken bazı sorunlar da bulunmaktadır:

- Düzlem yüzeyli hacimsel nesnelere yanında daha karmaşık geometrik yapıların modellenmesi için silindirik yüzeyli 3B nesnelere için de veri tipleri VTYS'de desteklenmelidir.
- R-Tree indeks modeli veritabanında depolanan 3B nesnelerin hızlı bir şekilde sorgulanmasını sağlayabilmektedir. Bunun yanında, yüksek performans sağlayacak yeni algoritma ve yapılar kullanılarak farklı konumsal indeksleme modellerinin geliştirilmesi gereklidir.

- VTYS bünyesinde, özellikle CBS uygulamalarının gerektirdiği analizleri gerçekleştirmek için konumsal ve öznetelik bilgilerini işleyebilecek çok sayıda konumsal fonksiyon ve operatör bulunmaktadır. 3B nesnelerin geometrik işlemleri (iki 3B nesnenin kesişimi sonucunda oluşan yeni nesnenin elde edilmesi), topolojik işlemleri (iki 3B nesnenin kesişiminin Boolean doğru/yanlış sonucunu döndüren) ve metrik işlemleri (3B nesnelerin hacimlerinin hesaplanması gibi matematik hesaplamalar) için daha fazla konumsal fonksiyona gereksinim vardır.
- İlişkisel VTYS içinde tutulan konumsal verilere ancak SQL komutları ile erişilebilmektedir. SQL komutlarının çalıştırılması sonucunda elde edilen sonuç ürün karakter ve sayılardan oluşmaktadır. 3B nesnelerin görsel ortamda sorgulanması ve düzenlenmesi işlemleri için veya sorgulama sonuçlarının görüntülenmesinde VTYS'de depolanan konumsal bilgiye çeşitli CBS/CAD araçlarıyla erişim gereklidir. CBS/CAD uygulamalarının hacimsel 3B veri tiplerinin temsiliyi desteklemesi gerekmektedir.

Teşekkür

Bu çalışma, Karadeniz Teknik Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından desteklenmiştir (Proje No: 2008.112.006.1). Çalışma kapsamında Delft Teknik Üniversitesi'nde (Hollanda) gerçekleştirilen araştırmaya katkılarından dolayı Prof. Peter van OOSTEROM ve Dr. Sisi ZLATANOVA'ya teşekkür ederiz.

Kaynaklar

- APEL M.: **From 3D Geomodelling Systems Towards 3D Geoscience Information Systems: Data Model, Query Functionality, and Data Management**, Computers & Geosciences 32, s.222–229, 2006, Elsevier.
- ARENS C., STOTER J.E. ve van OOSTEROM P.J.M.: **Modelling 3D Spatial Objects in a Geo-DBMS Using a 3D Primitive**, Computers & Geosciences, 31, s.165 – 177, 2005, Elsevier.
- BITENC M., DAHLBERG K., DONER F., van GOOR B., LIN K., YIN Y., YUAN X. ve ZLATANOVA S.: **Utility registration: Slovenia, China, Sweden and Turkey**, GIS Report No. 49, s.48, 2008 (<http://www.gdmc.nl/punlications> adresinden erişilebilir)
- DONER F., THOMPSON R., STOTER J., LEMMEN C., PLOEGER H. ve van OOSTEROM P.: **4D Land Administration Solutions in the Context of the Spatial Information Infrastructure**, FIG Working Week 2008, 14 – 19 June 2008, Stockholm, Sweden.
- DÖNER F. ve BIYIK C.: **Üç Boyutlu Kadastro**, Hkm - Jeodezi, Jeoinformasyon ve Arazi Yönetimi Dergisi, Sayı: 97, ISSN 1300-3534, s.53, 2007, HKMO Ankara.

- DU Y., ZLATANOVA S. ve LIU X.: **Management and 3D Visualisation of Pipeline Networks Using DBMS and AEC Software**, ISPRS Commission IV Symposium on Geospatial Databases for Sustainable Development, s. 27 – 30, September, 2006, Goa, India.
- KARATAŞ K.: **Kentsel Teknik Altyapı Tesisleri, Kadastro ve Türkiye'deki Uygulamaların Organizasyonu**, Doktora Tezi, KTÜ, Fen Bilimleri Enstitüsü, Mayıs 2007, Trabzon.
- KAZAR B.M., KOTHURI R., van OOSTEROM P. ve RAVADA S.: **On Valid and Invalid Three-Dimensional Geometries**, P. van Oosterom, S. Zlatanova, F. Penninga and E. Fendel (Editörler); *Advances in 3D Geoinformation Systems*, Springer, s.19 – 46, 2008.
- KHUAN C.T., ABDUL-RAHMAN A. ve ZLATANOVA S.: **3D Solids and Their Management In DBMS**, P. van Oosterom, S. Zlatanova, F. Penninga and E. Fendel (Editörler); *Advances in 3D Geoinformation Systems*, Springer, s. 279 – 311, 2008.
- KOLBE T., GROEGER G. ve CZERWINSKI A.: **City Geography Markup Language (CityGML)**, In: OGC, OpenGIS Consortium, Discussion Papers, Version 0.3.0, 2006.
- KOTHURI R., GODFRIND A. ve BEINAT E.: **Pro Oracle Spatial for Oracle Database 11g**, Apress, ISBN: 1590598997, 2007.
- LAN H., DEREK MARTIN C. ve LIM C.H.: **Rockfall Analyst: A GIS Extension for Three-Dimensional and Spatially Distributed Rockfall Hazard Modeling**, Computers & Geosciences, Volume 33, Issue 2, s.262 – 279, 2007, Elsevier.
- LEE J. ve ZLATANOVA S.: **A 3D Data Model and Topological Analyses for Emergency Response in Urban Areas**, In: S. Zlatanova and J. Li (Eds.); *Geospatial Information Technology for Emergency Response*, Taylor & Francis/Balkema, s.143 – 168, 2008.
- LEMMEN C. ve van OOSTEROM P.: **3D Cadastres**, Computers, Environment and Urban Systems 27, s: 337–343, 2003, Elsevier.
- ORACLE: **Oracle Spatial Developer's Guide, 11g**, Release 1 (11.1) Part No.B28400-02, 2007.
- SCHNEIDER J.P. ve EBERLY D.H.: **Geometric Tools for Computer Graphics**, Morgan Kaufman, 2002.
- STOTER J. ve ZLATANOVA S.: **Visualisation and Editing of 3D Objects Organised in a DBMS**, Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering, 22-24 January 2003, Enschede, The Netherlands.
- STOTER J., KLUIJVER H. ve KURAKULA V.: **3D Noise Mapping in Urban Areas**, International Journal of Geographical Information Science, Vol. 22, No. 8, s.907–924, 2008, Elsevier.
- TSE R., GOLD C. ve KIDNER D.: **An Original Way of Building a TIN with Complex Urban Structures**. In: Proceedings of ISPRS 2004 - XXth Congress, Istanbul, Turkey, 2004.
- XUL., GENG G., SHIM. ve LIN S.: **Pipe Network 3D Visualization Service Architecture**, in 2008 IEEE Congress on Services, Part I, s.495 – 502, 2008.
- ZLATANOVA S., RAHMAN A. ve PILOUK M.: **3D GIS: Current Status and Perspectives**, ISPRS Commission IV, WG IV/1, July 2002, Ottawa, Canada.
- ZLATANOVA S.: **3D GIS for Urban Development**, PhD thesis, ITC, The Netherlands, 2000.