

Web Tabanlı Uygulamaların Coğrafi Bilgi Sistemleri ile Desteklenmesi

Merve ÖZKAN*, İsmail Rakıp KARAŞ², Caner ÖZCAN³

¹Bilgisayar Mühendisliği / Lisansüstü Eğitim Enstitüsü, Karabük Üniversitesi, Türkiye

²Bilgisayar Mühendisliği / Mühendislik Fakültesi, Karabük Üniversitesi, Türkiye

³Yazılım Mühendisliği / Mühendislik Fakültesi, Karabük Üniversitesi, Türkiye

*(merveozkan867@gmail.com)

(Geliş Tarihi: 21 Ağustos 2023, Kabul Tarihi: 28 Ağustos 2023)

(1st International Conference on Recent and Innovative Results in Engineering and Technology ICRIRET 2023, August 16-18, 2023)

ATIF/REFERENCE: Özkan, M., Karaş, İ. R. & Özcan C. (2023). Web Tabanlı Uygulamaların Coğrafi Bilgi Sistemleri ile Desteklenmesi. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(7), 171-179.

Özet – Coğrafi Bilgi Sistemleri (CBS), nesnelerin konumlarını ve öznitelik bilgilerini belirli bir amaca yönelik toplama, bilgisayar ortamında depolama, güncelleştirme, kontrol etme, analiz etme ve görüntüleme gibi işlemlere olanak sağlayan karar destek sistemleridir. Web tabanlı uygulama geliştiricileri CBS'nin bu özelliğinden faydalanarak eldeki verileri daha da anlamlı hale getirebilir ya da yanlış yapılan analizlerin farkına varabilirler. Bu çalışma web tabanlı uygulamaların CBS ile desteklenmesi gerektiğini savunmakta ve geniş bir literatür taramasının ardından bu işlem için örnek bir senaryo içermektedir. Çalışmada veri tabanı olarak PostgreSQL, PostgreSQL'de jeo formatı destekleyen PostGIS eklentisi, veri dönüşüm formatı olarak GeoJSON, bağılıkların eklenmesi için Spring Boot altyapısı, Spring Boot ile Maven projesinin oluşturulması, projenin kodlanması için Java dili, harita desteği için Mapbox GL JS tercih edilmiştir. Çalışmanın sonunda Türkiye haritasında yer alan iller nüfus yoğunluğuna bağlı olarak temsili renklendirilmiştir.

Anahtar Kelimeler – Web Tabanlı CBS, GeoJSON, Mapbox GL JS, PostgreSQL, PostGIS, Harita Desteği

I. GİRİŞ

Coğrafi Bilgi Sistemleri (CBS), yeryüzüne ait nesnelerin konumlarını ve öznitelik bilgilerini belirli bir amaca yönelik toplama, bilgisayar ortamında depolama, güncelleştirme, kontrol etme, analiz etme ve görüntüleme gibi işlemlere olanak sağlayan karar destek sistemleridir [1]. CBS alanında oldukça hızlı büyüyen veri miktarı ve bilgi teknolojilerinde meydana gelen gelişmeler ile web tabanlı ve mobil uygulamalar tercih sebebi olmaya devam etmektedir [2]. Ayrıca web tabanlı uygulamalar ve mobil uygulamalar ile sisteme hızlı ve kolay erişilebildiğinden son kullanıcının CBS

ile etkileşimi artmaktadır. Web tabanlı CBS'ler sağladıkları olanaklar sebebiyle programcının isteğine göre yeniden şekillendirilebilirler [3]. Bu ve bunun gibi sebeplerden dolayı web tabanlı CBS'ler oldukça yaygınlaşmıştır.

Web tabanlı CBS denildiğinde son zamanlarda üç boyutlu (3B) yapılar ön plana çıkmaktadır. Structure From Motion (SFM) iki boyutlu görüntülerin birleştirilmesinden 3B yapıları oluşturabilmek için geliştirilen algoritmadır. Fotogrametrik görüntü eşleştirme işlemi yapan bu algoritma yaygın olarak kullanılmakta olan yeni nesil bulut tabanlı görüntü eşleme yazılımlarının

genel işlevidir [4]. Sefercik ve arkadaşları yeni nesil görüntü eşleme yazılımlarından VisualSFM, AgiSoft ve Pix4D SFM kullanarak 3B dijital yüzey modelleri (DYM) üretmişlerdir. Bu yazılımların artılarını ve eksilerini detaylı olarak değerlendirmişler. Elde ettikleri bulgulara göre Agisoft, Pix4D SFM ve VisualSFM yazılımlarından görsel performans açısından daha üstündür. Agisoft yüksek modelleme performansı elde edebilmek için daha fazla hava fotoğrafına ihtiyaç duymaktadır. Ayrıca Pix4D SFM'in oluşturduğu DYM ile Agisoft uyum göstermektedir [4]. Hamal ve Ulvi, Mersin iline ait Yenişehir ilçesinde bulunan Mersin Üniversitesi kampüsüne ait İnsansız Hava Aracı (İHA) görüntüleri üzerinde web tabanlı bir yazılım oluşturmuşlardır. Bu alandaki yapılar farklı geometrik özellik sergilemektedir. Bu çalışmada kullanılan İHA RGB bantlara sahip 21 megapiksel dahili kamera içermektedir. 230 dakikada 312 fotoğraf alınmıştır. Çekilen görüntülerin doğruluğunu test etme ve gerçek konum oluşturabilme amaçlı araziye 16 adet yer kontrol noktası kurulmuştur. Fotogrametri alanı sayesinde objenin çeşitli açılardan çekilmiş farklı görüntülerinden yararlanılır ve hassas bir sonuç elde edilir. Bu çalışma literatürde öne çıkan AgiSoft SFM tabanlı yazılımın güvenilir olduğundan hareketle AgiSoft'u kullanmıştır. Hamal ve Ulvi elde edilen fotoğrafları dengelemiş ve nokta bulutu halindeki verileri CBS yazılımına aktarmışlardır [5]. Yapılan diğer bir çalışmada ise üç farklı İHA sistemi vasıtasıyla ve üç farklı yöntemle yerleşim alanları yeniden oluşturulmuş ve her yöntemin avantaj ve dezavantajları detaylarıyla tartışılmıştır [6].

Web Graphics Library (WebGL) web tarayıcılarında 3B görseller çizdirebilmek için kullanılan bir programlama arayüzüdür. Bu modeller dinamik olarak Hyper Text Markup Language (HTML) sayfalarının içerisinde oluşturulur ve JavaScript kütüphanelerine entegre edilir [7]. WebGL teknolojileri arkeolojik alan keşfinden [8], tıbbi simülasyona [9] ve sel simülasyonuna [10] kadar birçok araştırma alanında kullanılmaktadır. Kullanıcılara görselleştirilmiş ortamla gerçek zamanlı etkileşim sunarak karmaşık 3B ortamlarda çevrimiçi gezinmeye olanak tanır [11]. Bu alandaki gelişmeler sayesinde web üzerinde karmaşık 3B modellerin oluşturulması, yeni çözümlerin üretilmesi ve test edilmesi olasılığı ile sanal gerçeklik alanında yeni fırsatlar oluşmuştur [12].

Bu sayede karmaşık 3B bir ortamda, cihaza herhangi bir uygulama yüklemeyen kullanıcılar tarafından çevrimiçi olarak gezilebilir. Bu çözümde 3B ortam oluşturmak için video oyunu teknolojileri kullanılır [13].

Son zamanlarda, coğrafi ve coğrafi olmayan verileri birçok Resource Description Framework (RDF) verilerine dönüştürmek için birçok RDF veri oluşturma aracı geliştirilmiştir. Nishanbaev ve arkadaşları mevcut son teknoloji jeo-uzamsal RDF oluşturma araçlarının ve birbirine bağlanan çerçevelerin özelliklerini ve işlevlerini karşılaştırmalı olarak değerlendirmişlerdir [14]. Buna ek olarak, makale jeo-uzamsal kültürel miras RDF verilerini oluşturmak ve bunları ilgili RDF verileriyle ilişkilendirmek için bir metodoloji sunmaktadır. Bu metodoloji bir CIDOC Kavramsal Referans Modeli ontolojisi kullanır ve RDF verilerini DBpedia ile birbirine bağlar. Bu metodoloji, kültürel miras araştırmacıları ve profesyonelleri için geliştirilmiş olsa da, diğer alan uzmanları tarafından da kullanılabilir. Geospatial Semantic Web, jeo-uzamsal verileri ve semantic web kavramlarını birleştirir. Jeo-uzamsal veriler, geometri gibi farklı özelliklere sahiptir. Geospatial Semantic Web'de jeo uzamsal verileri ifade etmek, depolamak ve sorgulamak için gerekli yapıları barındırır. Semantic web alanında kullanılan sorgulama dili SPARQL iken GeoSpatial semantic web'de GeoSPARQL olmaktadır [14].

Bulut bilişim bilgisayarlar ve diğer cihazlarda kullanıcılar arasında paylaşılan bilgisayar kaynakları sağlayan, internet tabanlı bilişim hizmetleridir [15]. Her alanda olduğu gibi web tabanlı CBS alanında da sunucusuz mimariler, gün geçtikçe daha fazla tanınır ve tercih edilir hale gelmiştir [16]. Sunucusuz mimariler; Hizmet olarak Yazılım (Software as a Service, SaaS), Hizmet olarak Platform (Platform as a Service, PaaS), Hizmet olarak Altyapı (Infrastructure as a Service, IaaS) şeklinde üç farklı kategori şeklinde ele alınmaktadır. Bu hizmet türlerinde donanım kullanımı son kullanıcıdan soyutlanarak gerçekleştirilmektedir [17]. Böylece alt yapı yönetimi konusunda bilinmesi gereken hem teknik bilgiler hemde alt yapı yönetimini sağlayacak olan insan kaynağı büyük ölçüde azaltılmış olur. Bu teknolojinin CBS alanında kullanılması bulut bilişim konusunda çok fazla bilgisi olmayan CBS uzmanlarını cesaretlendirmiştir [18].

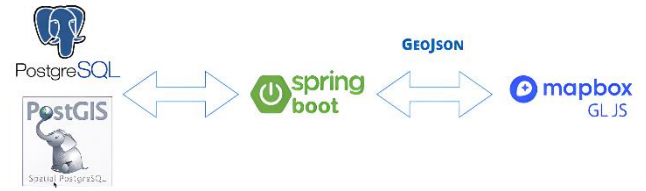
CityGML bir şehri gösterebilmek için daha küçük parçalar bölerek temsil eder. 3B kent modellerinin depolanması için oluşturulmuş bir Genişletilebilir İşaretleme Dilidir (XML). 3B CBS uygulamalarında oldukça fazla kullanılmaktadır [19]. Tapu ve Kadastro Genel Müdürlüğü (TKGM) 3B Kadastro Projesi kapsamında CityGML'i kullanmışlardır ve veri formatından ortaya çıkabilecek sorunları ele almışlardır [20]. CityGML oldukça popüler kullanılan bir veri formatıdır fakat buna rağmen pek çok eksiklik barındırmaktadır. Hiyerarşik veri modeli yaklaşımını benimsediğinden en alttaki verilere erişim yavaşken en üstteki verilere erişim hızlıdır, bu durum performansı olumsuz etkilemektedir. JavaScript Object Notation (JSON) yerine XML veri tipinde olması harici kütüphane kullanımını zorunlu kılmaktadır. Nesnelerin geometrik, semantik ve özniteliklerini iç içe barındırması diğer bir dezavantajı olarak saptamıştır. Bu ve bunun gibi diğer eksiklikler ele alınarak Usta ve arkadaşları tarafından CityGML veri formatının eksiklerini giderecek yeni bir CityJSON veri modeli önerilmiştir [20].

3B kent modelleri yüksek veri hacmine sahip modellerdir bu da web tabanlı olarak görüntülenmesi ve geliştirilmesi esnasında performans sorunlarına sebebiyet vermektedir [21], [22]. Literatürde 3B verilerin 3B Tiles spesifikasyonuna göre bölümleyebilen açık kaynak kodlu bir yazılım yer almamaktadır. Usta ve Cömert literatürdeki bu eksikliği R-Tree ve Adaptive QuadTree hiyerarşik veri yapılarını kullanıp açık kaynak kodlu bir yazılım geliştirerek gidermişlerdir [23].

1854 yılında kolera Londra'yı vurduğunda, birçok kişi hastalığın hava yoluyla yayıldığını düşünmüştür. İngiliz Dr. John Snow salgın yerlerini, yolları, mülk sınırlarını ve su hatlarını haritalamaya başlamıştır ve bu özellikler haritaya eklendiğinde kolera vakalarının genellikle su hattı boyunca bulunduğunu görmüştür. Ana sokaklardan birindeki su pompasını yerinden çıkararak salgını önlemiştir [24]. Uygulamalara CBS desteğinin eklenmesi ile elde edilen verilere yönelik 1854 kolera salgınına olduğu gibi yanlış çıkarımların önüne geçilmesi ve ek analiz sonuçları elde edilebilir. Bu çalışma web tabanlı uygulama geliştiricilere uygulamalarına harita desteği eklenmesinin faydaları konusunda ısrarcı olmakta ve nasıl yapılacağına dair örnek bir senaryo içermektedir.

II. MATERYAL VE YÖNTEM

Bu çalışma kapsamında veri tabanı olarak PostgreSQL, PostgreSQL veri tabanının konumsal verileri tanıyabilmesi için PostGIS (Post Geographical Information System) eklentisi, Rest (Representational State Transfer) Servisler için Spring Framework, sunucudan gelen yanıt tipi aşamasında GeoJSON (Geo Javascript Object Notation), elde edilen GeoJSON yapısının görüntülenmesi aşamasında ise Mapbox GL JS kullanılmıştır. Programlama dili olarak Java tercih edilmiştir. Şekil 1'de bahsedilen yapılara yer verilmiştir.



Şekil 1. Kullanılan teknolojiler

Çalışma kapsamında tercih edilen teknolojiler Şekil 1'de yer almaktadır. Kullanılan teknolojiler araştırmacının isteğine göre yeniden şekillendirilebilir. Alternatiflerine ilerleyen bölümlerde yer verilmiştir.

Bu çalışma kapsamında amaç web tabanlı uygulamalara harita desteği eklenmesi konusunda araştırmacıları teşvik etmek ve onlara örnek bir uygulama ile bakış açısı kazandırmak olarak hedeflenmiştir.

A. Kullanılan Teknolojiler

PostgreSQL

PostgreSQL 1977 yılında geliştirilmeye başlanmış, 1996 yılında ise Structured Query Language (SQL) standartlarını desteklemeye başlaması ile PostgreSQL olarak isimlendirilmiştir. Açık kaynak kodlu ilişkisel veri tabanı sistemidir. Bir veri tabanının hızlı olmasından ziyade güvenilir olması önem arz etmektedir. PostgreSQL performans ve güvenlik açısından gelişmiş düzeydedir. Ayrıca, ücretsiz ve platform bağımsızdır. İşlemler PostgreSQL ile birlikte kurulan pgAdmin arayüzünden yürütülmektedir. Bu çalışmada PostgreSQL'in 15.2.2 versiyonu kullanılmıştır. Bu versiyonla uyumlu sürüm pgAdmin 4'tür. Yerine kullanılacak olan alternatiflere örnek olarak Oracle, MySQL, MsSQL, MariaDB, MongoDB verilebilir.

PostGIS

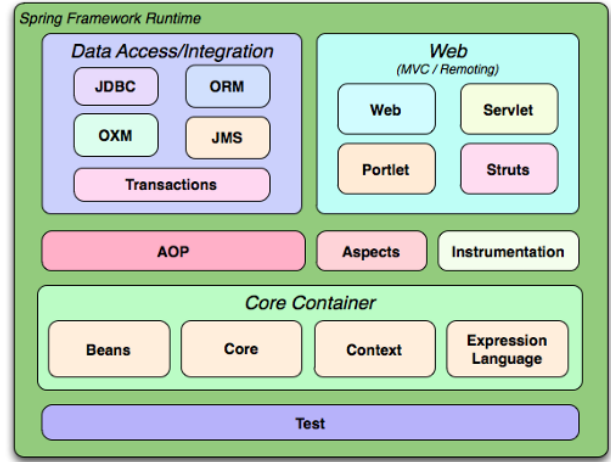
CBS nesnelere veri tabanında depolanmasına destek sağlayan PostgreSQL veri tabanı uzantısıdır. CBS nesnelere analizi ve işlenmesi için fonksiyonlar içermektedir. PostgreSQL kurulumundan sonra Application Stack Builder aracılığıyla uzantının veri tabanına eklenmesi sağlanmaktadır. Oluşturulan veri tabanında extensions sekmesinden PostGIS'in oluşturularak dahil edilmesi gerekmektedir. Bu eklenti sonrasında PostgreSQL'de geometry veri tipindeki yapılan haritalar üzerinde gösterildiği görülecektir.

PostGIS ile PostgreSQL'in sunduğu geometrik veri tipi karıştırılmamalıdır. PostgreSQL bünyesinde çok güçlü bir coğrafi veri tabanı modülü olan PostGIS uzantısını bulundurmaktadır. Bu uzantı konum bilgisi içeren verilerin saklanması için ötesinde coğrafi işlevleri sayesinde konumsal analizlerin de yapılabilmesini sağlayan ilave özellikler barındırır.

PostgreSQL veri tabanına PostGIS eklentisi kurulduktan sonra geometry veri tipinin geldiği görülecektir. Bunun haricinde PostGIS box2d, box3d, geometry, geometry_dump, geography veri tiplerini desteklemektedir. Box2d 2 boyutlu sınırlayıcı kutucuğu, box3d 3 boyutlu sınırlayıcı kutucuğu, geometry_dump karmaşık geometrinin parçalarını tanımlamak için kullanılan bileşik bir türü ifade etmektedir. Geography elipsoidal koordinat sistemi kullanırken geometry düzlemsel koordinat sistemi kullanmaktadır. Geography kullanılırsa geometry veri tipinde kullanılan tüm fonksiyonlar kullanılamaz. Proje başında kesin bir karar vermek bu noktada gereklidir [25].

Spring Framework

Spring Framework Java ile geliştirme yapmayı kolaylaştıran, yaklaşık 20 modülden oluşan açık kaynaklı bir yapıdır. Java uygulama geliştirme süreçlerini basitleştirmeyi hedeflemektedir. Şekil 2'de Spring Framework'ün yapısı yer almaktadır. Data Access/Integration, Web, AOP, Aspects, Instrumentation, Core Container ve Test olarak gruplandırıldığı görülmektedir [26]. Spring Boot Spring Framework ile geliştirilen, uygulama yapılandırılmasında yaşanan karmaşıklığı gidermek üzerine kuruludur.



Şekil 2. Spring Framework Yapısı (Johnson ve diğerleri, 2012)

Spring Initializr [27] kullanılarak inşa aracı, Java Sanal Makinesi (JVM) dilini, Spring Boot sürümünü, paketleme türünü, Java sürümünü, projede kullanılacak olan kütüphaneleri dinamik olarak belirlemeyi sağlar.

Bu çalışmada inşa aracı olan Maven, Spring Boot sürümü olarak 3.0.5, paketleme türü olarak jar, JVM sürümü olarak 11, kütüphaneler ise PostgreSQL Driver, Spring Web, Spring Data JDBC olarak belirlenmiştir. Spring initializr arayüzünde ilgili ayarlar oluşturularak bir Proje nesne dosyası, pom.xml elde edilmiştir. Pom, xml dosya formatında olan ve bir projenin inşa edilmesinden, bağımlılıklarından ve paketleme yöntemine kadar birçok bilgiyi içeriğinde barındıran bir dosyadır.

GeoJSON

Ağustos 2016 tarihinde İnternet Mühendisliği Görev Gücü tarafından yayımlanan RFC 7946 standardı 2008'de belirlenen standardın yerini almıştır ve bu standart hala GeoJSON yapısı için kullanımdadır [28]. Basit coğrafi özellikleri ve bunların uzamsal olmayan niteliklerini temsil etmek için tasarlanmış JSON formatıdır. Bir GeoJSON nesnesi türü Feature nesne ya da Feature Collection nesne olabilir. Geometry veri tipi öklid koordinat sistemine ait verileri tutmak için kullanılmaktadır. GeoJSON'da geometry nesnesi türü Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon ve FeatureCollection olabilir. Şekil 3 nokta, Şekil 4 çizgi, Şekil 5 poligon içeren GeoJSON dosya yapılarını içermektedir.

```

{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [35, 39]
  },
  "properties": {
    "adi": "Şehir Merkezi"
  }
}

```

Şekil 3. Nokta geometry tipi içeren geoJSON formatı

```

{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [35, 39],
      [36, 40],
      [37, 38],
      [38, 39]
    ]
  },
  "properties": {
    "tipi": "Nehir",
    "adi": "Kızılırmak"
  }
}

```

Şekil 4. Çizgi geometry tipi içeren geoJSON formatı

```

{
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [35, 39],
        [37, 38],
        [36, 40],
        [35, 39]
      ]
    ]
  },
  "properties": {
    "tipi": "Şehir",
    "adi": "Gaziantep"
  }
}

```

Şekil 5. Poligon geometry tipi içeren geoJSON formatı

Şekil 3'te nokta tabanlı geoJSON formatı x ve y koordinatlarına ait bilgiden meydana gelirken, Şekil 4'te çizgi tabanlı geoJSON formatı koordinatlar dizisinden meydana geldiği görülür. Şekil 5'te tipi poligon olan geometry nesnesinin koordinatlarında birden fazla enlem, boylam ikilisinin olduğu görülmektedir. Poligon'un, çizgi geometri tipinden farkı; poligon kapalı bir alan olduğundan başlangıç ve bitiş koordinatlarının aynı olması gerekliliğidir. Nesnenin geometrik şekli ve coğrafi konumu kaydedildikten sonra şekli tanımlayan özelliklerin ise properties anahtarı alanına kaydedilmesi gerekmektedir. Bu çalışmada nokta, çizgi ve poligon geometri veri tipleri üzerinde işlemler gerçekleştirilmiştir. Ayrıca coğrafi nesnelerin tutulabildiği dosya türleri gml ve kml uzantılı dosyalar olarak örneklendirilebilir.

Mapbox GL JS

Web tabanlı harita kütüphaneleri coğrafi verileri gösterebilmek ve bu verilerle işlem yapabilmek için Uygulama Programlama Arayüzü (API) kullanır. Coğrafi verileri tarayıcı üzerinden haritada göstermek için birçok JavaScript kütüphanesi yer almaktadır. Bunlardan bazıları; ArcGIS API, Mapbox GL JS, OpenLayers, Leaflet'dir [29]. Mapbox GL JS, görsel bir görünüme sahip olmanın yanı sıra, haritalara etkileşimli özellikler eklemek ve veri odaklı görselleştirmeler oluşturmak için bu çalışmada tercih edilmiştir.

Java

Çalışma java dilini kullanarak çözüm geliştirmiştir. NetBeans IDE'sinin 12.6 versiyonu kullanılmıştır. JDK 12.0.2 sürümü tercih edilmiştir.

B. Veri

İnternette üç adet backup dosyası elde edilmiştir. Bu tablolar içerisindeki veriler nokta, çizgi ve poligon olarak yer almaktadır. Nokta verilerinin yer aldığı tabloda Türkiye'de bulunan bazı merkezlerin koordinatları geometry veri tipiyle gösterilmektedir ve 5000 adet satırdan oluşan tablodur. Tablo 1'de nokta verilerine ait tablo yapısına ve örnek veriye yer verilmiştir.

Tablo 1. Nokta verilerinin yer aldığı tablodan örnek

id[PK] numeri	Tipi character varying (250)	il character varying (250)	ilce character varying (250)	Geom geometry
254884 7	Banka ATM	Ankara	Akyurt	01010000AE61 ...
736367	Resmi Kurum	Çorum	Dodurga	01010000A0E6 1...

Diğer tabloda çizgi olarak gösterilebilen yapılar Türkiye'deki nehirlerdir. Çizgi veri tipleri birden fazla koordinat verisinin bir araya gelmesiyle elde edilmektedir. Bu sebepten Tablo 1'de yer alan geom sütunundan çok daha fazla büyüklükte string'den oluşmaktadır. Tablo 2'de çizgi verilerine ait tablo yapısı ve örnek veri yer almaktadır.

Tablo 2. Çizgi verilerinin yer aldığı tablodan örnek veri

id[PK] numeric	adi character varying (250)	Geom geometry
1	Aras N.	01O2000020E61...
2	Aras N.	01010000A0E61...

Tablo 3'te poligon verilerine ait tablo yapısı ve örnek veriler yer almaktadır. Bu tabloda Türkiye'deki iller ve illere ait nüfus bilgileri yer almaktadır. Nüfus verileri temsilidir.

Tablo 3. Poligon verilerinin yer aldığı tablodan örnek veri

id[PK] numeric	il character varying (250)	nüfus numeric	Geom geometry
41	Konya	2232374	01O2000020E61...
6	Ankara	5639076	01010000A0E61...

Geometri Veri Oluşturma

Bir lokasyon bazlı uygulama geliştirilmek istendiğinde kullanıcıdan konum bilgisi alınır ve veri tabanına enlem ve boylam bilgileri şeklinde farklı sütunlara kayıt edilebilir. Ama verileri bu şekilde kaydetmek PostGIS'in sunduğu birçok analiz fonksiyonunu çalıştırmaya engel olacaktır. Bu sebepten geometry tipi konumsal verilerin analiz tarafında kullanılmalıdır.

Web tabanlı uygulamaları CBS ile desteklemek istediğimizde elimizde geometry veri tipi hali hazırda bulunmayacaktır. Bu sebepten bu başlık altında geometry veri tipinin PostgreSQL'de nasıl oluşturulacağıyla alakalı detaylara yer verilmiştir.

Poligon tanımlarken en az üç adet koordinat verisine ihtiyaç vardır. Bir koordinat verisinden diğerine gitmeli sonrasında başladığı noktaya geri dönmelidir. Fakat burada yapılan milimetrik koordinat tanımlama hataları bile oluşturulan verinin geometry verisine ait olmasını engellemektedir. Bu dikkat edilmesi gereken önemli konulardan biridir.

Geometry oluştururken PostgreSQL'de tanımlı fonksiyonlar kullanılabilir. Geometry verileri Tablo 1, 2 ve 3'te yer aldığı gibi string görünümlüdür.

```
SELECT st_geomfromtext('POINT (32.86186, 39.9090847)', 4326)
```

```
SELECT st_geomfromjson('{"type" : "Point", "coordinates" : [32.7703187, 39.9449081]}')
```

Yukarıdaki kod satırında koordinatları verilen yer Ankara Mamak yakınlarında bir noktadır. Enlem, boylam bilgisi st_geomfromtext fonksiyonuna verilmiştir. 4326 sayısı noktanın coğrafi projeksiyonda nokta olduğunu göstermektedir. Zorunlu kullanım değildir fakat bu şekilde yapılması bir geometri sütunu oluştururken kısıtlama girmekte fayda olduğundan önerilmektedir. Kod bloklarında yer alan her iki satır çalıştırıldığında sonucu string olan bir geometry verisinin üretildiği gözlemlenmiştir. Koordinatlar GeoJSON tipinde geldiyse st_geomfromgeojson fonksiyonu kullanılmalıdır. Birçok geometry veri tipi oluşturma yöntemi vardır. Mantık aynıdır fakat girdi değerleri farklıdır. Çizgi tipinde geometry oluşturmak için aşağıda yer alan kod bloğu bu işlemi gerçekleştirecektir. St_makeline fonksiyonu içerisinde 2 ya da daha fazla geometri olarak çizgileri oluşturacaktır.

```
SELECT st_makeline(
st_geomfromtext('POINT(32.8618697,39.9090847)',4326) ,
st_geomfromtext('POINT(33.86186,39.9090847)',4326)
)
```

Bu sorgular sonucunda string elde edilmiştir ama verinin tipinin geometry olduğu görülmüştür. PostGIS eklentisinin geometry viewer sekmesinden, girilen koordinatlara ait görsellerin harita üzerine konumlandırılacağı görülecektir.

C. Projenin Kodlanması

Pom.xml uzantılı maven projesi oluşturulduktan sonra Java'da veri tabanı bağlantısının oluşturulması gerekir. Bu projede Java ile PostgreSQL veri tabanı arasındaki bağlantı aşağıdaki kod bloğunda görüldüğü gibi oluşturulmuştur.

```
spring.datasource.url = jdbc:postgresql://localhost/postgresql
spring.datasource.username = postgresql
```

```
spring.datasource.password = *****
server.port = 5432
server.servlet.context-path = /geojsonserver
```

Bağlantı sağlandıktan sonra veri tabanına sorgudan gelen parametreler GeoJSON formatına dönüştürülecek ve Mapbox GL JS harita desteğiyle ekranda gösterilecektir. Bu işlem için MapBox'un kendi sitesinden üyelik oluşturulmuş ve üye sayfasında yer alan API, Java projesinde kullanılan JavaScript kodları arasına eklenmiştir. Mapbox GL JS her üyeye farklı API kodu üreterek işlem yapmaktadır ve ücretsizdir. Şekil 6'da nokta verilerinin harita üzerindeki görüntüsüne; Şekil 7'de çizgi verilerinin harita üzerindeki görüntüsüne yer verilmiştir.



Şekil 6. Nokta verilerinin harita üzerinde gösterilmesi



Şekil 7. Çizgi verilerinin harita üzerindeki görüntüsü

Poligon verisi Türkiye il sınırlarından oluşmaktadır. Öznitelik olarak her ilin nüfus sayısı yer almaktadır. Poligonun özniteliklerine göre renklendirme yapıldığında Şekil 8'de yer alan görsel ortaya çıkmıştır.



Şekil 8. Poligon verisinde yer alan temsili nüfus rakamlarına göre haritanın renklendirilmesi

Renklendirmede 500000 kadar olan nüfuslu illerde '#00FF00' renk kodu, 1000000'a kadar olan değerde '#FFFF00' renk kodu, 3000000'a kadar olan illerde '#FFA500' renk kodu, 50000000'na kadar olan illerde '#FF0000' renk kodu, 50000000 ve üzerinde '#8B0000' renk kodu kullanılmıştır.

III. SONUÇLAR

1854 yılında kolera Londra'yı vurmuştur. Bu dönemde birçok kişi hastalığın hava yoluyla yayıldığını düşünmüştür. İngiliz Dr. John Snow salgın yerlerini, yolları, mülk sınırlarını ve su hatlarını haritalamaya başlamıştır ve bu özellikler haritaya eklendiğinde kolera vakalarının genellikle su hattı boyunca bulunduğunu görmüştür. Ana sokaklardan birindeki su pompasını yerinden çıkararak salgını önlemiştir [24]. Uygulamalara harita desteğinin eklenmesi ile 1854 kolera salgınında olduğu gibi yanlış çıkarımların önüne geçilmesi sağlanabilir ve böylece daha detaylı analiz sonuçları elde edilebilir. Muhasebe alanında uygulama geliştiren programcılar uygulamalarına harita desteği ekleyerek faturalarını konum bilgisi ile destekleyerek müşterilerinin hangi bölgelerde yoğunlaştığını daha net görebilirler. Bu ve bunun gibi örnekler daha da çoğaltılabilir. Bu çalışma web tabanlı uygulama geliştiricilere uygulamalarının CBS ile desteklenmesi konusunda bir bakış açısı kazandırmayı hedeflemekte ve bunu örnek bir senaryo ile desteklemektedir.

PostgreSQL'de veri tabanı işlemleri ile tablolarda yer alan bilgi koordinat bilgileri çekilmiştir. PostgreSQL'de harita desteği olan POSTGIS eklentisi tercih edilmiştir. Bu sayede konum verileri veri tabanı ortamında da görsel olarak anlamlı hale gelmiştir. Veri tabanından gelen veriler GeoJSON veri dönüşüm formatı kullanılarak JAVA ortamında kullanılmaya hazır hale getirilmiştir. Bağlılıkların ve kütüphane uyumluluklarının sağlanması için Spring Boot altyapısı terci edilmiştir. Spring Boot ile Maven projesi oluşturulmuş ve gerekli çalışma ortamının temeli atılmıştır. Proje JAVA dili kullanılarak kodlanmış ve harita desteği için MapBox GL JS tercih edilmiştir. Çalışmanın sonunda Türkiye haritasında yer alan iller nüfus yoğunluğuna bağlı olarak temsili renklendirilmiştir.

KAYNAKLAR

- [1] M. Taşyürek and M. Çelik, "RNN-GWR: A geographically weighted regression approach for frequently updated data," *Neurocomputing*, vol. 399,

- pp. 258–270, Jul. 2020, doi: 10.1016/J.NEUCOM.2020.02.058.
- [2] M. Taşyürek and M. Çelik, “FastGTWR: A Fast Geographically and Temporally Weighted Regression Approach,” *J. Fac. Eng. Archit. Gazi Univ.*, vol. 36, no. 2, pp. 715–726, 2021, doi: 10.17341/gazimmfd.757131.
- [3] İ. Yalçın, B. Anbaroğlu, and S. Kocaman, “Açık Kaynaklı Web Tabanlı Coğrafi Bilgi Sistemi Geliştirilmesi,” no. August, 2018, doi: 10.15659/uzalcbcs2018.6195.
- [4] U. G. Sefercik, F. Tanrıku, and C. Atalay, “SFM Tabanlı Yeni Nesil Görüntü Eşleştirme Yazılımlarının Fotogrametrik 3B Modelleme potansiyelerinin karşılaştırılması,” *Türkiye Fotogram. Derg.*, vol. 2, no. 2, 2020.
- [5] S. N. G. Hamal and A. Ulvi, “3B Kent Modelleri Oluşturma Sürecinde İHA Fotogrametrisi ve CBS Entegrasyonu: Mersin Üniversitesi Çiftlikköy Kampüsü Örneği,” *Türkiye Coğrafi Bilgi Sist. Derg.*, vol. 4, no. 2, 2022, doi: 10.56130/tucbis.1208096.
- [6] M. Yakar and Y. Dogan, “3D Reconstruction of Residential Areas with SFM Photogrammetry,” *Adv. Sci. Technol. Innov.*, pp. 73–75, 2019, doi: 10.1007/978-3-030-01440-7_18/COVER.
- [7] M. La Guardia, M. Koeva, F. D’Ippolito, and S. Karam, “3D Data Integration for Web Based Open Source Webgl Interactive Visualisation,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 48, no. 4/W4-2022, pp. 89–94, 2022, doi: 10.5194/isprs-archives-XLVIII-4-W4-2022-89-2022.
- [8] D. Tanasi, D. Fredrick, N. Reynolds, K. Kingsland, B. Jenkins, and S. Hassam, “In Ersilia ’ s Footsteps,” 2021.
- [9] Q. Min, Z. Wang, and N. Liu, “An Evaluation of HTML5 and WebGL for Medical Imaging Applications,” *J. Healthc. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/1592821.
- [10] C. E. Kilsedar, F. Fissore, F. Pirotti, and M. A. Brovelli, “Extraction and Visualization of 3D Building Models in Urban Areas for Flood Simulation,” *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 42, no. 2/W11, pp. 669–673, 2019, doi: 10.5194/isprs-archives-XLVII-2-W11-669-2019.
- [11] J. Huixian, “The Construction of Virtual Simulation Platform for Pingtan Experimental Area Based on HTML5 and WebGL,” <https://doi.org/10.1080/17517575.2019.1683766>, vol. 14, no. 9–10, pp. 1457–1474, 2019, doi: 10.1080/17517575.2019.1683766.
- [12] G. Bian and B. Wang, “Advances in Human Factors in Wearable Technologies and Game Design,” vol. 973, 2020, doi: 10.1007/978-3-030-20476-1.
- [13] S. K. Yıldız and G. Bozkurt, “Sanal Gerçekliğin Yeni Anakarası: Metaverse,” *TRT Akad.*, vol. 8, pp. 268–293, 2023, doi: 10.37679/trta.1203353.
- [14] I. Nishanbaev, E. Champion, and D. A. McMeekin, “Article a Comparative Evaluation of Geospatial Semantic Web Frameworks for Cultural Heritage,” *Heritage*, vol. 3, no. 3, pp. 875–890, 2020, doi: 10.3390/heritage3030048.
- [15] Chowhan Kuldeep, *Hands-On Serverless Computing: Build, run and orchestrate serverless ... - Kuldeep Chowhan - Google Libros*. 2018. Accessed: Apr. 16, 2023. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=YzhmDwAAQBAJ&oi=fnd&pg=PP1&dq=Google+cloud+function+history&ots=NuRj2_Vttk&sig=J2wWhInDt4cpWSPInzSTczHoR5Y#v=onepage&q&f=false
- [16] I. Baldini *et al.*, “Serverless Computing: Current Trends and Open Problems,” *Res. Adv. Cloud Comput.*, pp. 1–20, 2017, doi: 10.1007/978-981-10-5026-8_1.
- [17] S. Anand, A. Johnson, P. Mathikshara, and R. Karthik, “Real-time GPS tracking using serverless architecture and ARM processor,” *2019 11th Int. Conf. Commun. Syst. Networks, COMSNETS 2019*, no. January, pp. 541–543, 2019, doi: 10.1109/COMSNETS.2019.8711273.
- [18] M. E. Pakdil and R. N. Çelik, “Bulut Bilişimde Sunucusuz Mimariler ile Coğrafi Bilgi Teknolojilerinin Kullanımı Üzerine Bir İnceleme,” *J. Geod. Geoinf.*, vol. 10, no. 1, pp. 1–15, 2023, doi: 10.9733/jgg.2023r0001.t.
- [19] J. Zhu, G. Wright, J. Wang, and X. Wang, “A critical review of the integration of geographic information system and building information modelling at the data level,” *ISPRS Int. J. Geo-Information*, vol. 7, no. 2, pp. 1–16, 2018, doi: 10.3390/ijgi7020066.
- [20] Z. Usta, A. T. Akın, and Ç. Cömert, “TKGM CityGML Veri Modelinin 3B WebCBS Uygulamaları Açısından Investigation of the TKGM CityGML Data Model in the Context of 3D WebGIS Applications,” *Turkish J. Remote Sens. GIS*, vol. 3, no. 1, pp. 65–75, 2022.
- [21] D. Akkaya and M. Başaraner, “Yapı Bilgi Modellemesi ve Coğrafi Bilgi Sistemleri,” *Akad. Bilişim ’12 - XIV. Akad. Bilişim Konf.*, no. 2008, pp. 477–483, 2012.
- [22] X. Liu, X. Wang, G. Wright, J. C. P. Cheng, X. Li, and R. Liu, “A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS),” *ISPRS Int. J. Geo-Information*, vol. 6, no. 2, pp. 1–21, 2017, doi: 10.3390/ijgi6020053.
- [23] Z. Usta and Ç. Cömert, “Üç Boyutlu Konumsal Verinin Web Tabanlı Yönetiminde , OGC Standardı 3D Tiles ’ ın Hiyerarşik Veri Yapılarıyla Gerçekleştirimi,” *Turkish J. Remote Sens. GIS*, vol. 3, no. 1, pp. 76–85, 2022.
- [24] J. C. (James C. Thomas, D. J. (David J. Weber, and I. Ovid Technologies, *Epidemiologic methods for the study of infectious diseases*. Oxford University Press, 2001. Accessed: Apr. 16, 2023. [Online]. Available: <https://global.oup.com/academic/product/epidemiologic-methods-for-the-study-of-infectious-diseases-9780195121124>
- [25] “About PostGIS | PostGIS.” <https://postgis.net/> (accessed Apr. 15, 2023).
- [26] R. Johnson *et al.*, “Spring Framework Reference Documentation,” *Spring Framew.*, p. 819, 2012.
- [27] “Spring Initializr.” <https://start.spring.io/> (accessed Apr. 15, 2023).
- [28] “GeoJSON.” <https://geojson.org/> (accessed Apr. 16, 2023).

- [29] M. O. Mete and T. Yomralıođlu, “Açık Kaynaklı Bulut CBS Yardımıyla Kitlemel Taşınmaz Deđerleme Uygulaması,” *Harit. Derg.*, vol. 165, pp. 28–42, 2021, [Online]. Available: <https://www.harita.gov.tr/images/dergi/makaleler/49bc2d462a0890.pdf>