Getir CBS Uzamsal Veritabanı Proje Raporu

# Proje Adı:

Getir Tabanlı CBS Veritabanı

# Hazırlayan:

Berkay Akyolal

# Veritabanı Adı:

getir\_db

# Kullanılan Teknolojiler:

PostgreSQL + PostGIS, Python (GeoPandas, SQLAlchemy, Matplotlib, GeoPy (Nominatim), Shapely(Point))

## 1. Proje Amacı

## 2. Veritabanı Tasarımı ve Veri Aktarımı

- Taban Harita Verisi: OpenStreetMap (.osm) uzantılı XML dosyası  
- Aktarım Aracı: osm2pgsql  
- Çekilen Temel Katmanlar:  
 - planet\_osm\_point: POI ve yerler  
 - planet\_osm\_line: Yollar  
 - planet\_osm\_polygon: Bina ve alan geometrileri, Köprüler  
  
Veriler PostgreSQL + PostGIS ortamında UTF-8 kodlama ile çekildi. Tablolarda PostGIS üzerinden uzamsal sorgulara uygun indekslemeler ve geometri alanları hazır hale getirildi.

## 3. Gerçekleştirilen Uzamsal Sorgular

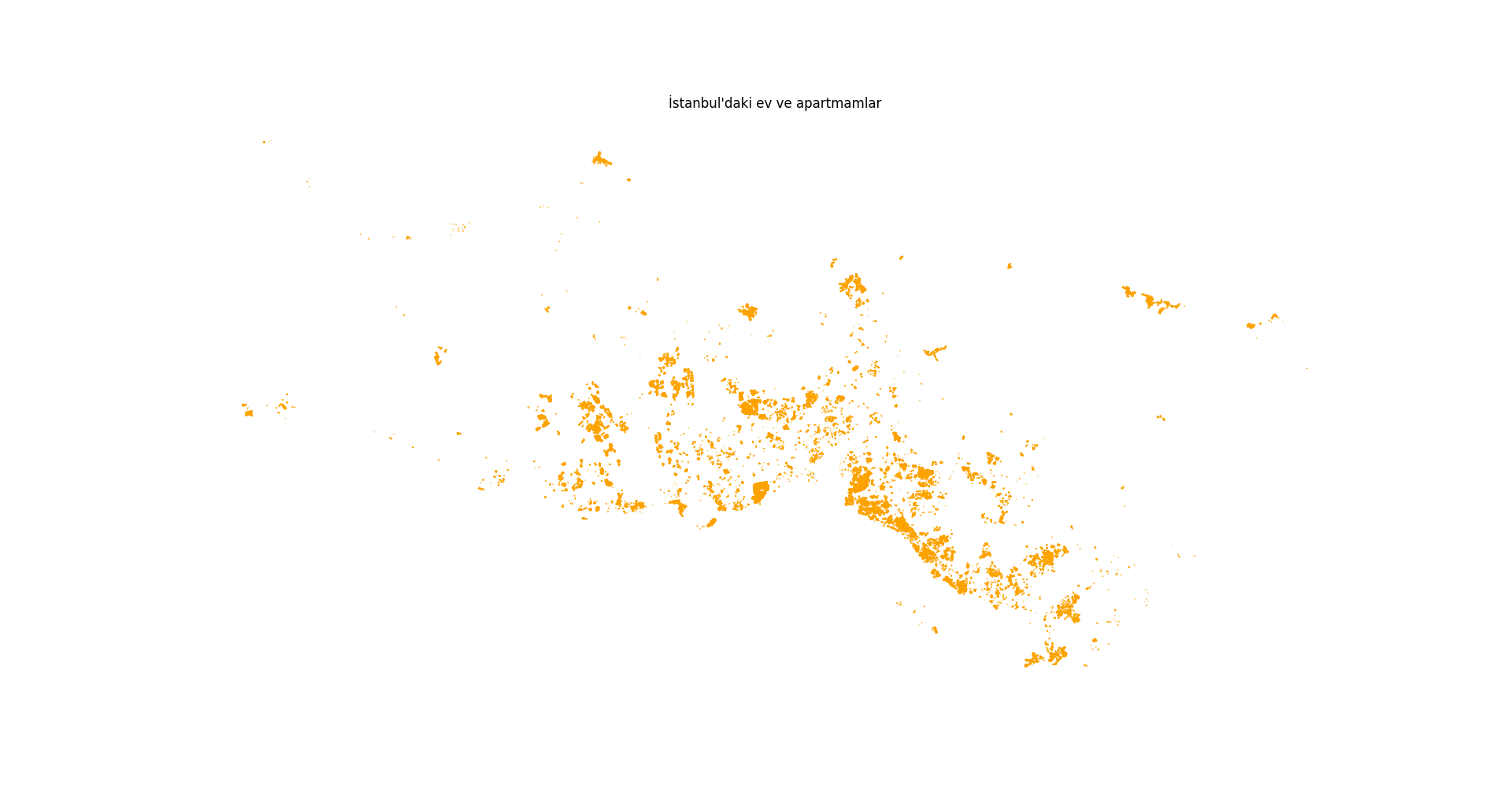
### 3.1 Ev ve Apartmanları Listeleme

SQL:  
select name, building, way

from planet\_osm\_polygon

where building in ('house','apartments');

### Amaç: İstanbul’daki evleri ve apartmanları çıkarıp temel bir fikir oluşturmak. (Not: Ev veya apartman olarak işaretli veri sayısı gerçek değildir. Düzeltmeye çalışsam da yapamadım.)



### 3.2 Market ve Dağıtım Yerlerini Listeleme

SQL:  
SELECT name, shop, amenity, building, way

FROM planet\_osm\_point

WHERE shop IS NOT NULL

OR amenity IN ('cafe','food\_court','ice\_cream','marketplace','post\_office')

OR building IN ('apartments','hotel','hospital','house','office','school','university'

);

Amaç: Hem kullanıcı hem dağıtıcı firma için fikir vermesi için market ve dağıtım yerlerini listelemek.

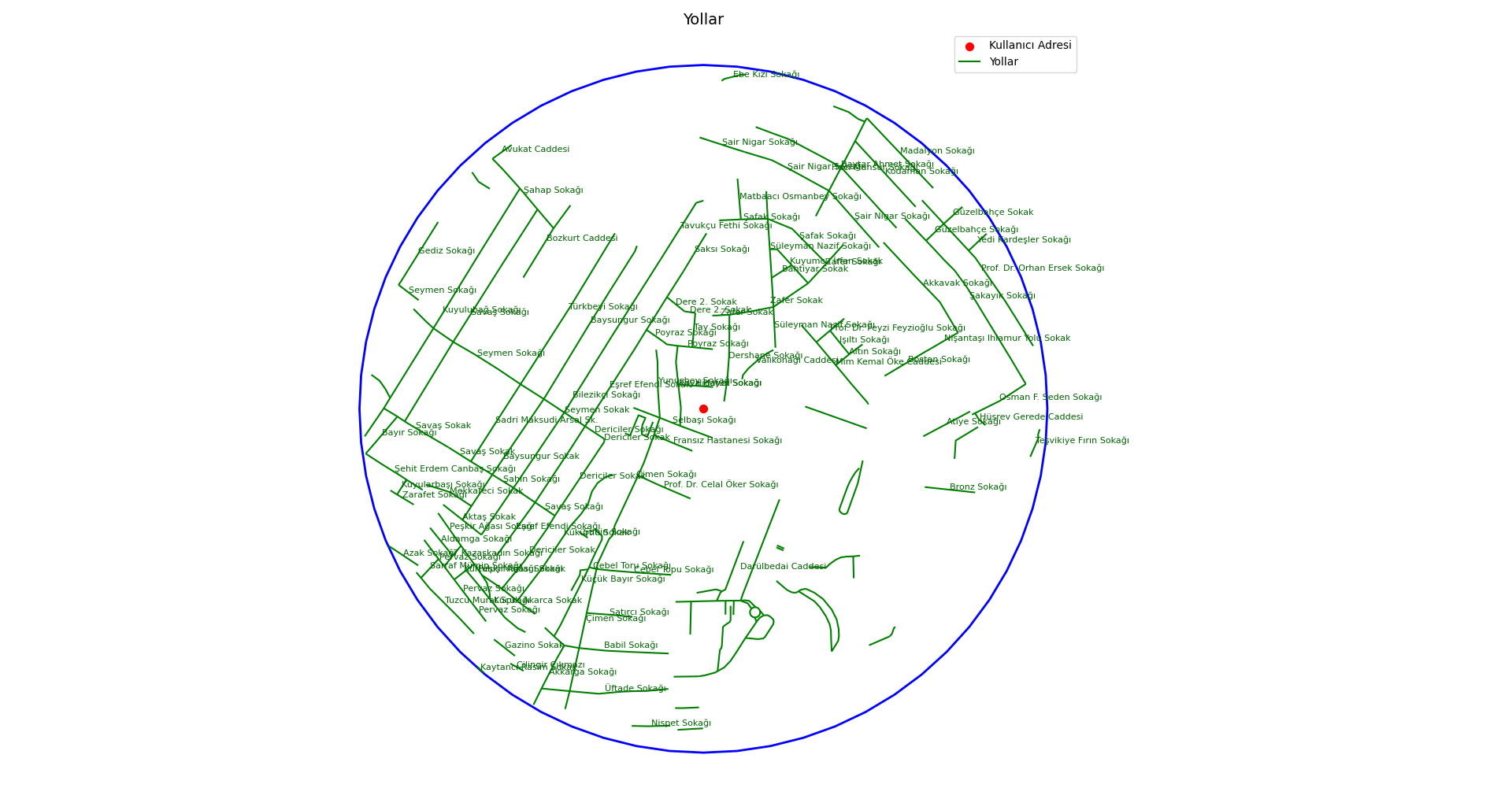
### 3.3 Gidilebilecek Yolları Listeleme

SQL:  
SELECT name, highway, bridge, way

       FROM planet\_osm\_line

        WHERE highway IN ('residential', 'service')

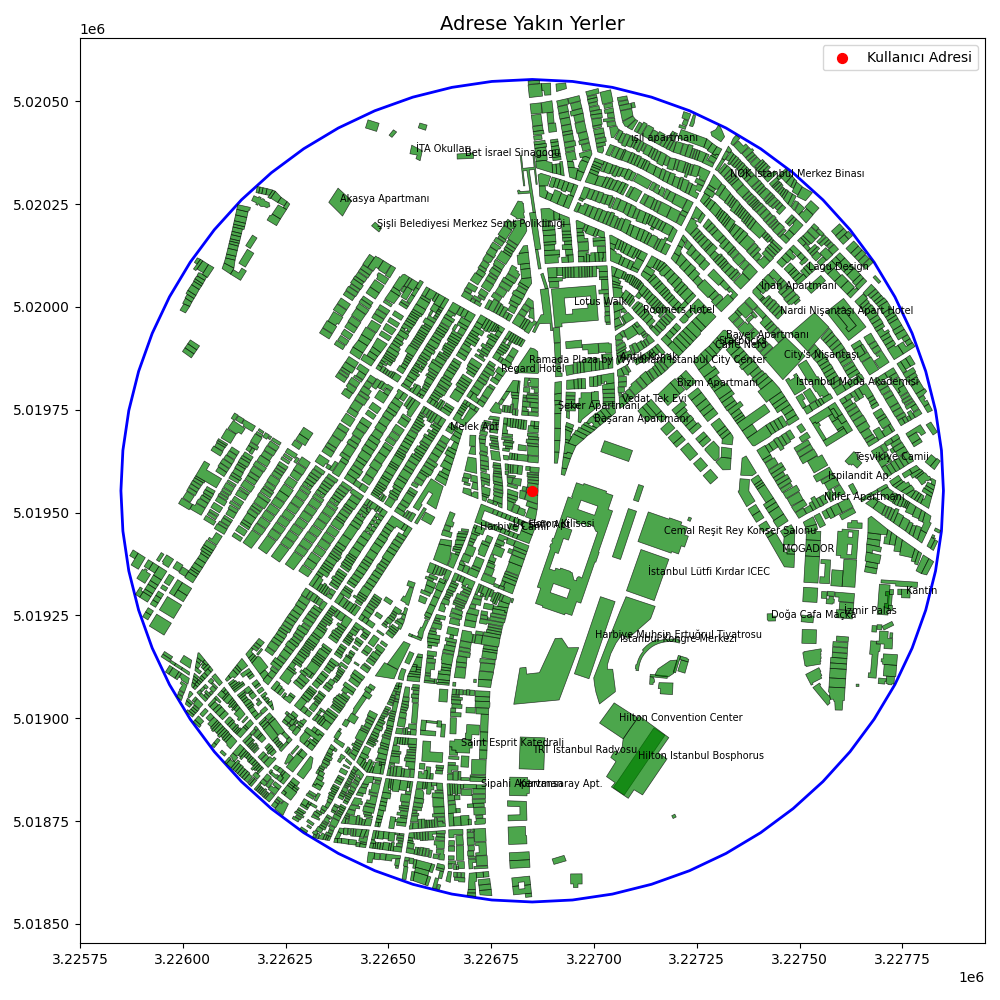
        OR bridge IS NOT NULL;

Amaç: Kuryeden kendi konumunu girmesini sağlayıp konumuna yakın yolları çıkarmak.

### 3.4 Tüm Binaları Listeleme

SQL:  
SELECT name, building, way  
FROM planet\_osm\_polygon  
WHERE building IS NOT NULL;

Amaç: İstanbul’daki tüm binaları listeleyerek kuryeler için gerekli verinin sağlanması ve kuryenin konum bilgisini kullanarak 1000m çapındaki binaları göstermek.



### 3.5 Adrese En Yakın 25 Market

SQL:  
SELECT name, shop, amenity, building, way

FROM planet\_osm\_point

WHERE shop IS NOT NULL

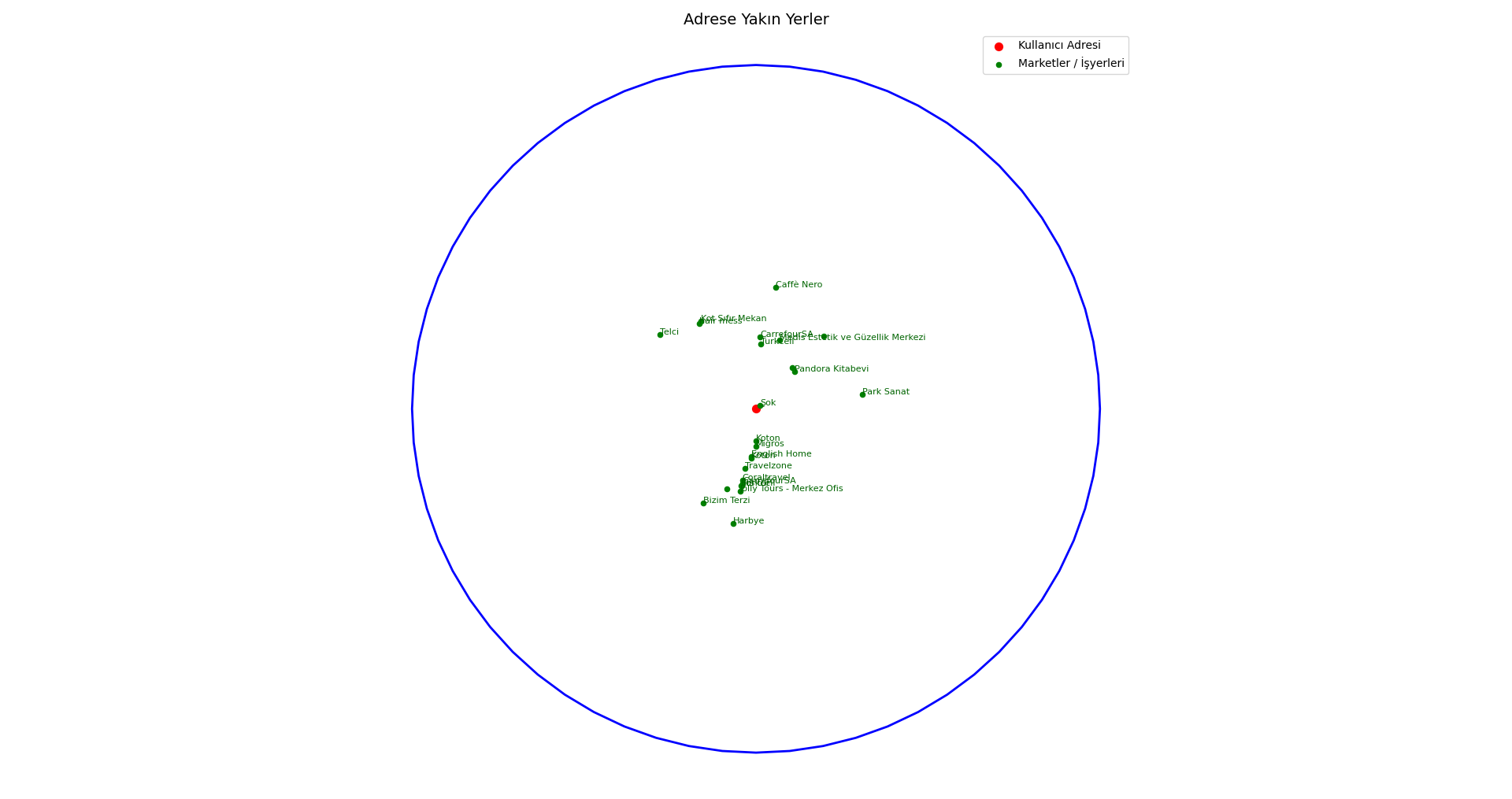
OR amenity IN ('cafe','food\_court','ice\_cream','marketplace','post\_office')

OR building IN ('apartments','hotel','hospital','house','office','school','university')

ORDER BY way <-> ST\_Transform(ST\_SetSRID(ST\_MakePoint(41.02052, 28.78869), 4326), 3857)

LIMIT 25;

Amaç: Kullanıcıdan adresini alıp kendisine en yakın konumdaki 25 marketi listemek.



## 4. Python ile Görselleştirme

Python tarafında GeoPandas kullanılarak SQL sorgularının görselleştirilmesi yapılmıştır. Ayrıca, GeoPy kullanılarak kullanıcı veya kuryeden girilecek adres girdisini koordinat verisine dönüştürülmesi sağlanmıştır. Sorgulara göre:

-Kullanıcın konumu kırmızı noktayla belirlenmiştir.

-Binalar, mağazalar ve yollar da yeşil olarak belirlenmiştir.

## 5. Değerlendirme

✔ Güçlü Yönler:  
- Gerçek verilerle çalışıldı  
- Uzamsal SQL yetkinliği geliştirildi  
- Python ile çok katmanlı görselleştirme başarıldı

⚠ Zorlanılan Noktalar:  
- Kodlama sürecinde şifre hataları ve eksik uzantılar

-Sorguları Python diline aktarma süreci

✘ Eksik Kalanlar:  
- Kullanıcı arayüzü ile etkileşim (ekstra opsiyonel olabilirdi, belki mobil üstünden kullanıcı açısından kullanımı olurdu.)

## 6. Sonuç

Proje, CBS ve uzamsal veritabanı alanlarında temel becerileri uygulamalı olarak geliştirmeyi başarmıştır. PostgreSQL + PostGIS ile çalışılması, SQL sorgularının harita verilerine uygulanması ve Python ile bu çıktılarının anlamlı görsellere dönüştürülmesi başarılı şekilde rapora yansımıştır.