

# NewMind AI Bootcamp 7. Dersin Ödevi

Hazırlayanın Adı-Soyadı: Yunus BİLGİÇ

**Proje Tanımı:** Teknik Destek Ekibi İçin Kurumsal Bilgi Grafi Tabanlı Akıllı Chatbot

**Proje Açıklaması:** Bu projede, teknik servis personelinin müşteri şikayetlerini daha hızlı çözebilmesi için **Neo4j graph veri tabanı** ve **LLM (Büyük Dil Modelleri)** destekli bir **yardımcı chatbot** tasarlanmıştır. Proje sayesinde çalışanlar doğru doküman yönetimi ile bilgiye hızla ulaşır. Ayrıca yeni gelen personel çok daha hızlı bir biçimde adapte olur. Grafik yapısı sayesinde şikayetler, arıza kodları, parçalar, çözüm önerileri, sistem bölümleri ve mağaza bilgileri birbiriyle ilişkilendirilebilir. Ayrıca **vector embedding** ile semantik aramalar da desteklenerek teknik personelin doğal dilde sorduğu sorulara akıllı yanıtlar verilebilir.

## Veri Setinin Özellikleri

Çalışmada kullanılan ürün modeli, hata kodları, manuel ve çözümleri Siemens markasının websitesinde bulunmuştur. Müşteri şikayetleri, kayıt numarası, müşterinin şehri gibi sütunlar için sentetik veriler üretilecektir.

Sütun Adı	Açıklama
ID	Kayıt numarası (birincil anahtar)
Customer Complaint	Müşterinin ifadesiyle sorunu
Problem	Teknik personelin sınıflandırdığı problem başlığı
Error Code	E18, E03 gibi hata kodları
Part	Sorunun ilgili olduğu yedek parça (örn. Drain Pump)
Compartment / System	Arızanın ait olduğu bölüm (örn. Tahliye sistemi, Motor)
Solution	Uygulanabilecek teknik çözüm
Manuel	Teknik PDF ya da doküman linki
Product Name / Model	İlgili ürün bilgisi (örn. Siemens WM3000)
Location	Müşterinin şehri / bölgesi
Store	Bölgedeki yetkili servis veya mağaza bilgisi (ad / telefon)

## Projenin Kullanım Alanları

Projede kullanılacak yapı kurumsal bilgi grafi şeklindedir. Bunun sayesinde kurumdaki birçok departman bu chatbottan yararlanabilir.

- **Teknik Servis Personeli** → Arıza tespiti, parça kontrolü ve çözüm adımları için kullanır.  
(Örneğin çamaşır makinası, klima arıza bakım personelleri)
- Yeni başlayan personellerin arıza ve çözümlere çok hızlı bir biçimde adapte olur ve orta uzmanlıkta olan personelin hızlı bir biçimde uzmanlaşır.
- Çalışanların arızanın çözümünü teknik manuel ve çözüm videolarına çok hızlı bir biçimde erişimi sağlanır.
- Geçmişte zor bir biçimde çözülmüş arızanın bilgisi kurum hafızasında kalmış olur, böylece tekrar eden veya benzer durumlarda ek bilgi sağlanmış olur.
- **Çağrı Merkezi Elemanları** → Müşteri açıklamalarından sorunu tespit etme ve çözüme ulaştırma aşamasında oldukça yardımcı olur.
- **Depo Sorumluları** → Gerekli parça stoklarını anında görebilir.
- **Yönetim** → Sık tekrarlayan arızaları ve çözüm performanslarını analiz edip ARGE ekiplerine bildirimini sağlayabilir.
- Farklı departmanlar arasında iletişim sağlanmış olur.
- Şirketin normal şartlarda görülmeyen ilişkilerinin tespiti ve geliştirmelerin kolaylıkla yapılması

## Graph Yapısındaki Node (Düğüm) Türleri ve Özellikleri

### 1. Customer: Müşteri

- Id: Müşteri id'si

### 2. City: Şehir

- name: Şehrin adı

### 3. Store: Müşterinin bulunduğu konumda hizmet veren teknik servis mağazası.

- name: Mağaza adı
- phone: Telefon numarası

### 4. CustomerComplaint: Müşteri şikayet metni ve benzersiz ID bilgisi.

- id: Şikayet ID'si
- text: Müşteri tarafından bildirilen şikayet metni
- **textEmbedding: Text vektor embeddingi**

### 5. Problem: Müşteri şikayetini ve bunun teknik anlamda karşılığını temsil eder.

- description: Problemin teknik tanımı
- **descriptionEmbedding: Description vektor embeddingi**

### 6. ErrorCode: Üründe meydana gelen probleme karşılık gelen hata kodudur.

- code: Hata kodu (örnek: "E18")
- description: Hata açıklaması (örnek: "Drainage problem")

### 7. Solution: Belirli bir problemin çözümüne yönelik açıklamayı içerir.

- description: Çözüm adımı (örnek: "Filtreyi temizle")
- estimated\_time: Tahmini onarım süresi
- difficulty: Zorluk seviyesi (örnek: Easy, Medium)

### 8. Product: Sorunun yaşandığı ürün ya da cihaz.

- brand: Marka (örnek: Siemens)
- model: Model (örnek: WM3000)

**9. System:** Problemin ait olduğu teknik sistem ya da bölüm.

- name: Sistem/Bölüm adı

**10. Part:** Arızanın çözümünde kullanılan yedek parça veya ekipmandır.

- name: Parçanın adı
- warehouse\_location: Depodaki yeri (örnek: "A3 - 007")
- stock: Stok miktarı

**11. Manuel:** Arıza çözümü için oluşturulmuş teknik dokümanlardır.

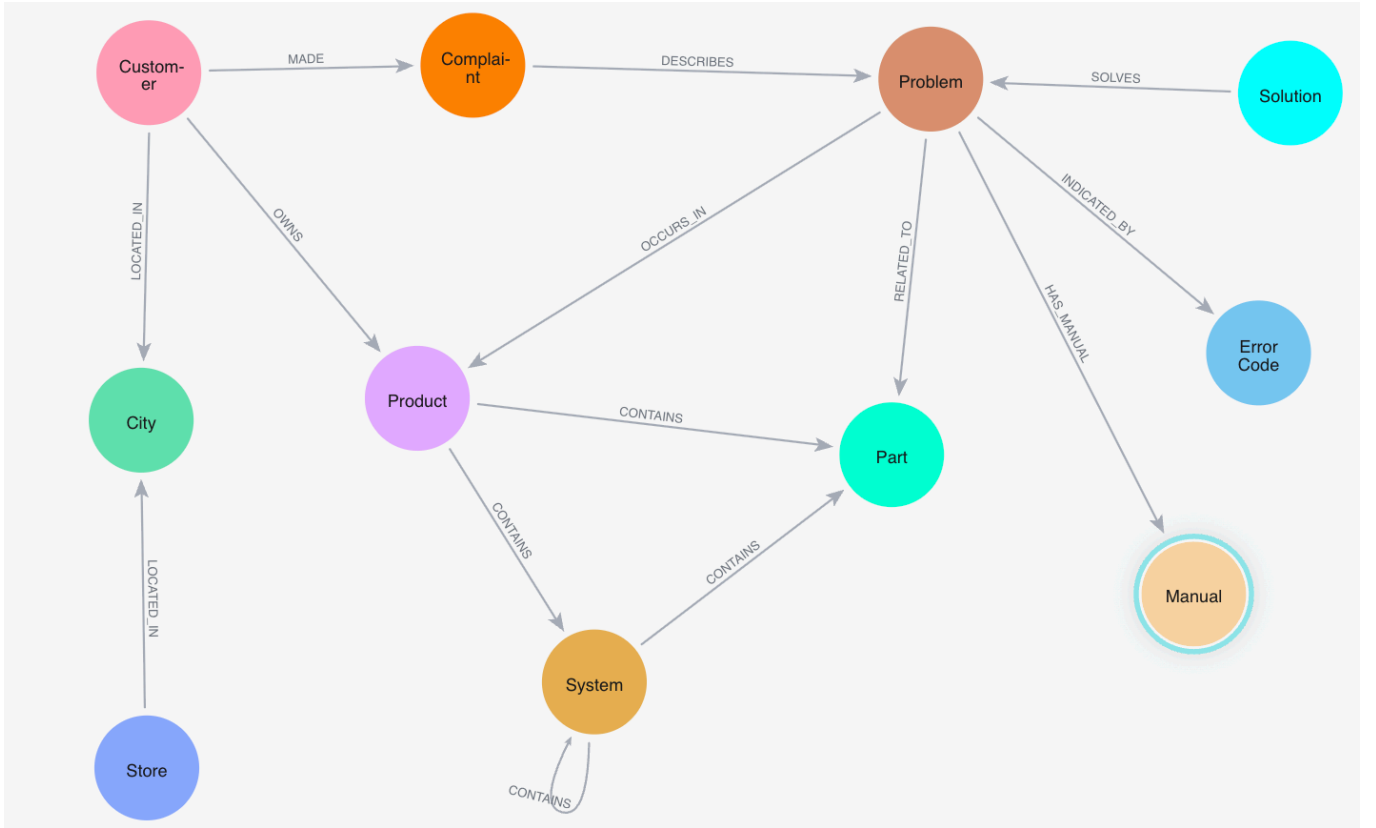
- Name: Manuelin adı veya id'si

### Graph Yapısındaki İlişkiler (Relationships)

- (Customer)-[:OWNS]->(Product) :Müşterinin sahip olduğu ürünü belirtir.
- (Customer)-[:LIVES\_IN]->(City) :Müşterinin yaşadığı şehri belirtir.
- (City)-[:HAS\_STORE]->(Store) :Şehirde bulunan servis mağazalarını gösterir.
- (Customer)-[:MADE]->(Complaint): Müşterinin oluşturduğu şikayeti gösterir.
- (Complaint)-[:DESCRIBES]->(Problem): Şikayetin teknik heyet tarafındaki tanımını belirtir.
- (Problem)-[:HAS\_ERROR\_CODE]->(ErrorCode): Problemin sahip olduğu hata kodu gösterir.
- (Problem)-[:SOLVED\_BY]->(Solution): Problemin çözümünü gösterir.
- (Problem)-[:HAS\_MANUAL]->(Manual): Problem için oluşturulmuş teknik dokümanları gösterir.
- (Problem)-[:RELATED\_PART]->(Part): Problem ile ilişkili yedek parçaları gösterir.
- (Problem)-[:OCCURS\_IN]->(Product): Problemin meydana geldiği ürünü gösterir.
- (Part)-[:PART\_OF]->(System): Parçanın içinde bulunduğu sistemi gösterir.
- (Product)-[:CONTAINS]->(System): Ürünün sahip olduğu sistemleri gösterir.
- (System)-[:PART\_OF]->(System): Büyük bir sistemin içindeki alt sistemleri gösterir.

## Graph Şeması

CALL db.schema.visualization()



## Bu Proje Hangi Sorulara Cevap Verebilir?

1. "Makinem su boşaltmıyor" problemi hangi parçayı ve sistemi etkiler?
2. Müşterilerin en çok şikayet ettiği arıza nedir?
3. "Drain Problem" için önerilen çözümler nelerdir?
4. E18 hata kodu hangi arıza ile ilişkilidir?
5. "X12 Bar" parçası hangi sistem ve ürünle ilişkilidir?
6. En çok arıza çıkaran sistem ve parça hangisidir?
7. En çok arıza çıkaran ürün nedir?
8. Bir müşteri hangi şehirde yaşıyor ve ona en yakın mağaza hangisi?
9. Bir çözümün teknik dökümanı nelerdir?
10. Hangi parça hangi rafta bulunmaktadır?
11. Hangi arızalar arasında ilişki bulunmaktadır? (Bu bazı problemlerin sebebinin doğru yerde aranmasını sağlayabilir.)

## Bu Projede Graph Yapısının Kullanılmasının Avantajları Nelerdir?

### 1. İlişkisel Sorguların Doğal ve Hızlı Yapılması

**Avantaj:** Arıza, çözüm, parça, sistem ve ürün gibi kavramlar arasındaki çoklu bağlantılar karmaşık SQL JOIN'ler yerine tek adımda, doğal şekilde sorgulanabilir.

**Örnek:**

- Bir müşteri şikayetinden yola çıkarak ilgili problemi, parçayı, çözümü ve ürün modelini birkaç satırlık Cypher sorgusuyla bulabilirsiniz.
- Parça → Sistem → Ürün zinciri bir sorguda kolayca takip edilir.

### 2. LLM + Vector Embedding ile Semantik Aramaya Uygunluk

**Avantaj:** Müşteri şikayetleri veya problemleri vector embedding olarak node'a gömülerek semantik benzerlik aramaları yapılabilir.

**Sonuç:**

- "Makinem su boşaltmıyor" → "Drain Problem" gibi eşleşmeler vektör düzeyinde yapılır.
- Graph üzerinden hangi çözümün daha sık işe yaradığını da analiz edebilirsiniz.

### 3. Gerçek Hayattaki Yapının Doğal Modellemesi

**Avantaj:** Parçalar sistemlere, sistemler ürünlere aittir. Müşteriler şehirlerde yaşar, şehirlerde mağazalar vardır. Bu doğal hiyerarşi, graph yapısıyla birebir modellenir.

**Sonuç:**

- Model karmaşıklığı arttıkça (örneğin: ürün → sistem → parça → problem → çözüm zinciri) bile veritabanı mimarisi değişmeden gelişir.

### 4. Kolay Veri Keşfi ve Görselleştirme

**Avantaj:** Neo4j arayüzü veya Bloom gibi araçlarla veri görselleştirme kolaydır. Özellikle teknik destek ekipleri için hangi problemler daha sık görülüyor gibi analizler görsel olarak yapılabilir.

**Sonuç:**

- Teknik ekipler için görsel dashboard oluşturmak kolay.
- Hangi parçalar daha sık arızalanıyor gibi analizler birkaç tıklamayla yapılabilir.



## 5. Bağlantı Odaklı Analizler (Centrality, Clustering)

**Avantaj:** Graph teorisi uygulayarak hangi parçalar veya sistemler daha kritik (en çok bağlantılı) gibi analizler yapılabilir.

**Sonuç:**

- En sık arızalanan sistemleri bulma.
- En çok kullanılan çözüm adımlarını tespit etme.



## 6. Gelecekte Kural Motorları veya Öneri Sistemine Uygunluk

**Avantaj:** Graph üzerinde kural bazlı sistemler kolay uygulanabilir. Örn: “Şu üründe bu parça arızalanmışsa şu çözümü öner” gibi.

**Sonuç:**

- LLM olmadan bile node ilişkilerine dayalı öneri sistemleri kurabilirsiniz.



## 7. Teknik Destek veya Bakım Ekipleri İçin Doğrudan Uyum

**Avantaj:** Teknik ekip bir ürünle ilgili problemleri, ilgili parçaları ve çözüm geçmişini grafik olarak görebilir.