**Giriş**

**Domain Tanımı:**

Bu projede kitaplar, yazarlar ve kullanıcıların kitaplara verdiği puanlar arasındaki ilişkileri temsil eden bir graph veritabanı tasarladım. Amacım yazarların kitaplarının popülerliğini analiz etmek kullanıcıların tercihlerini anlamak ve kitap önerileri için temel oluşturabilecek bir graph modeli ortaya koymak.

Projede aşağıdaki temel **node** ve **relationship** yapıları yer aldı:

------------------------------------------------------------------------------------------------------------

| **Node** | **Property Örnekleri** |
| --- | --- |
| **Book** | title, ISBN, year, genre |
| **Author** | Name, birthYear, nationality |
| **User** | Name, userId |

---------------------------------------------------------------------------------------------------------------

| **Relationship** | **Property Örnekleri** |
| --- | --- |
| **WROTE** | – (Yazarın kitabı yazdığı ilişki) |
| **RATED** | Rating (puan), date |

metin, ekran görüntüsü, daire içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* Draw.io ile oluşturduğum nodelar ve propertyleri görselleştirmesi

**Cypher Sorguları ile Kitapları ve İlişkileri Ekleme**

**Uygulama**

Neo4j Aura üzerinde çalışırken, tüm kitapları ve yazarları tek bir sorguda topluca eklemiştim. Ancak, bu dökümanın daha açıklayıcı ve öğretici olması için her kitabı ve ilgili sorguları tek tek yazmaya karar verdim. Böylece yapının nasıl kurulduğu daha net görülebilecek.

Her kitap için şu üç adımı izledim:

* Kitap düğümünü oluşturmak.
* Yazar düğümünü oluşturmak.
* Yazar ile kitap arasındaki **WROTE** ilişkisini kurmak.

Ayrıca kullanıcılar ve onların kitaplara verdikleri puanlar için:

* Kullanıcı düğümünü oluşturmak.
* **RATED** ilişkisini kurmak.

**Onyx Storm – Rebecca Yarros**

**Cypher:**

MERGE (b:Book {ISBN: '978-1649377159'})

SET b.title = 'Onyx Storm',

b.year = 2025,

b.genre = ' Fantasy Romance';

MERGE (a:Author {name: 'Rebecca Yarros'})

SET a.birthYear = 1980,

a.nationality = 'ABD';

MERGE (a)-[:WROTE]->(b);

**Great Big Beautiful Life – Emily Henry**

MERGE (b:Book {ISBN: '978-0593441275'})

SET b.title = 'Great Big Beautiful Life',

b.year = 2025,

b.genre = ' Romantic Fiction';

MERGE (a:Author {name: 'Emily Henry'})

SET a.birthYear = 1985,

a.nationality = 'ABD';

MERGE (a)-[:WROTE]->(b);

**The Perfect Divorce – Jeneva Rose**

MERGE (b:Book {ISBN: '978-1662506133'})

SET b.title = 'The Perfect Divorce',

b.year = 2025,

b.genre = 'Thriller';

MERGE (a:Author {name: 'Jeneva Rose'})

SET a.birthYear = 1984,

a.nationality = 'ABD';

MERGE (a)-[:WROTE]->(b);

**Engereğin Gözü – Zülfü Livaneli**

MERGE (b:Book {ISBN: '978-9751045187'})

SET b.title = 'Engereğin Gözü',

b.year = 2024,

b.genre = 'Crime';

MERGE (a:Author {name: 'Zülfü Livaneli'})

SET a.birthYear = 1946,

a.nationality = 'Türkiye';

MERGE (a)-[:WROTE]->(b);

**Günübirlik Hayatlar – Irvin D. Yalom**

MERGE (b:Book {ISBN: '978-6052997850'})

SET b.title = ' Daylight Lives ',

b.year = 2024,

b.genre = 'Psychology';

MERGE (a:Author {name: 'Irvin D. Yalom'})

SET a.birthYear = 1931,

a.nationality = 'ABD';

MERGE (a)-[:WROTE]->(b);

**Gece Yarısı Kütüphanesi – Matt Haig**

MERGE (b:Book {ISBN: '978-6051981560'})

SET b.title = ' The Midnight Library ',

b.year = 2020,

b.genre = 'Fantasy';

MERGE (a:Author {name: 'Matt Haig'})

SET a.birthYear = 1975,

a.nationality = 'İngiltere';

MERGE (a)-[:WROTE]->(b);

**Kullanıcılar ve RATED İlişkileri:**

MERGE (u1:User {userId: '1'})

SET u1.name = 'Ayşe';

MERGE (u2:User {userId: '2'})

SET u2.name = 'Mehmet';

MERGE (u3:User {userId: '3'})

SET u3.name = 'Elif';

**Puanlama ilişkileri:**

MATCH (u:User {userId: '1'}), (b:Book {ISBN: '978-1649377159'})

MERGE (u)-[r:RATED]->(b)

SET r.rating = 5,

r.date = date('2025-05-04');

MATCH (u:User {userId: '2'}), (b:Book {ISBN: '978-0593441275'})

MERGE (u)-[r:RATED]->(b)

SET r.rating = 4,

r.date = date('2025-05-03');

MATCH (u:User {userId: '3'}), (b:Book {ISBN: '978-1662506133'})

MERGE (u)-[r:RATED]->(b)

SET r.rating = 5,

r.date = date('2025-05-02');

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Kitaplar, yazarlar, kullanıcılar ve aralarındaki ilişkileri tek bir Cypher sorgusu ile oluşturduğum komut. Bu sorgu sayesinde graph veritabanında kitapları ve yazarları tanımladım, her kitabın yazarı ile olan WROTE ilişkisini ve kullanıcıların kitaplara verdiği puanları RATED ilişkisiyle modelledim. Böylece veritabanımın temel domain yapısını oluşturmuş oldum.

ekran görüntüsü, daire içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Veriler arasındaki görünmeyen bağlantıları ortaya çıkaran graph modelimin görsel temsili. Kitap-yazar ilişkileri (WROTE) ve kullanıcıların kitaplara verdiği puanlar (RATED) ile birlikte

Daha küçük parçaları da incelemek istersek:

daire, ekran görüntüsü, astronomi, sanat içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Elif isimli kullanıcının(label: User) The Perfect Divorce isimli kitapla RATED ilişkisi olduğu ve aynı zamanda Jeneva Rose isimli yazarın da aynı kitapla WROTE ilişkisi olduğunu gözlemleyebiliyoruz.

daire, ekran görüntüsü, metin, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Graph Veritabanı Analizleri ve Sorguları**

Bu bölümde kitaplar, yazarlar ve kullanıcılar arasında kurduğum graph yapısı üzerinde yaptığım analizleri ve hazırladığım Cypher sorgularını paylaşıyorum. Amacım; bu graph modeli üzerinden yazarların popülerliğini analiz etmek, kullanıcıların okuma alışkanlıklarını anlamak, kitap önerileri geliştirmek ve genel olarak veriyi daha anlamlı hale getirmekti.

Her sorgunun farklı bir amacı var:  
Örneğin bazı sorgularla hangi yazarların daha popüler olduğunu görebiliyor, bazılarıyla da kullanıcıların hangi tür kitaplara ilgi gösterdiğini analiz ediyorum. Ayrıca kitapların zaman içinde aldığı puanları incelemek veya henüz keşfedilmemiş kitapları önermek için de graph’ın sunduğu bağlantısal yapıdan yararlanıyorum.

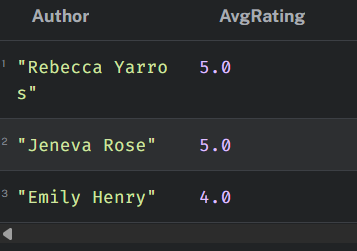
**Yazarların Popülerlik Analizi**

**Ortalama puana göre yazarları sıralama:**

MATCH (a:Author)-[:WROTE]->(b:Book)<-[r:RATED]-()

RETURN a.name AS Author, AVG(r.rating) AS AvgRating

ORDER BY AvgRating DESC;



**En çok kitap yazan yazarlar:**

MATCH (a:Author)-[:WROTE]->(b:Book)

RETURN a.name AS Author, COUNT(b) AS BookCount

ORDER BY BookCount DESC;

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Bu eklediğim veri yapısı çok küçük olduğu ve her yazarın sadece 1 kitabı olduğu için böyle bir sonuç beklemekteydim zaten. Eğer her yazarın daha çok kitabı eklenmiş olsaydı onları büyükten küçüğe sıralanmış halde gözlemleyecektik.

**Kullanıcı Tercih Analizi**

**Her kullanıcının en çok puanladığı tür:**

MATCH (u:User)-[:RATED]->(b:Book)

WITH u, b.genre AS Genre, COUNT(\*) AS Count

ORDER BY Count DESC

RETURN u.name AS User, Genre, Count;

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Yazar-Kitap Ağı Görselleştirme**

**Tüm yazar ve kitaplarını listele :**

MATCH (a:Author)-[:WROTE]->(b:Book)

RETURN a, b;

renklilik, daire, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Belirli bir yazarın kitapları (örnek: Rebecca Yarros):**

MATCH (a:Author {name: 'Rebecca Yarros'})-[:WROTE]->(b:Book)

RETURN a, b;

metin, ekran görüntüsü, daire, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Return ederken bana sadece nodeları döndürmesini istediğim için iki node’u gözlemliyoruz.

**Kitapların Puan Trendleri**

**Belirli bir kitabın puan tarihçesi (örnek: Onyx Storm):**

MATCH (b:Book {title: 'Onyx Storm'})<- [r:RATED]- (u:User)

RETURN u.name AS User, r.rating AS Rating, r.date AS Date

ORDER BY r.date ASC;

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**En Yüksek Puanlı Kitaplar**

**Kitapların ortalama puanına göre sıralama:**

MATCH (b:Book)<-[r:RATED]-()

RETURN b.title AS Book, AVG(r.rating) AS AvgRating, COUNT(r) AS RatingCount

ORDER BY AvgRating DESC, RatingCount DESC;

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Kitap Yayın Yılına Göre Trend Analizi**

**Yıllara göre ortalama puanları listele:**

MATCH (b:Book)<-[r:RATED]-()

RETURN b.year AS Year, AVG(r.rating) AS AvgRating, COUNT(r) AS RatingCount

ORDER BY Year ASC;

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Kullanıcılar Kaç Kitap Puanlamış?**

MATCH (u:User)-[:RATED]->(b:Book)

RETURN u.name AS User, COUNT(b) AS RatedBooks

ORDER BY RatedBooks DESC;

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Sonuç**

Bu projede kitaplar, yazarlar ve kullanıcılar arasındaki ilişkileri graph veritabanı üzerinde modelleyerek farklı analizler gerçekleştirdim. Hazırladığım Cypher sorguları sayesinde yazarların hangi kitapları yazdığı, kullanıcıların hangi kitapları ne kadar beğendiği ve genel olarak graph’taki bağlantılar daha görünür hale geldi.

Graph veritabanlarının en büyük avantajı; ilişkisel verileri çok doğal bir şekilde modellememizi sağlaması. Kitap önerileri oluştururken veya popülerlik analizleri yaparken, düğümler arasındaki bağlantıları kullanmak oldukça sezgisel ve güçlü sonuçlar veriyor.

Bu çalışma, bana hem Neo4j kullanımını hem de graph tabanlı düşünebilme becerimi geliştirme fırsatı sundu. İlerleyen aşamalarda bu temel yapı üzerine daha ileri seviye öneri sistemleri veya makine öğrenmesi entegrasyonları da eklemeyi planlıyorum.

Sonuç olarak; bu tür bir graph yapısı kitap öneri sistemleri, okuma trend analizleri ve hedef kitleyi daha iyi tanımak isteyen tüm platformlar için güçlü bir temel sunuyor.