NewMind AI Bootcamp 1.Hafta Ödevi

Adı-Soyadı: Yunus BİLGİÇ

# Bölüm 1: Generative AI (GenAI) Hakkında Kısa Bilgiler

Generative AI, verilen bir prompta karşılık yeni içerikler üretme yeteneği olan yapay zeka sistemleridir.

Metin, görsel, ses, video ve kod gibi içeriklerler üreten Generative AI modelleri bulunmaktadır. Metin üretmede ChatGPT ve DeepSeek, görsel üretmede DALL-E, ses üretiminde Siri ve Google Asistan, video üretmede Magisto ve SORA gibi modeller öne çıkmaktadır. Güvenliğe önem veren bireyler için Claude, araştırmacılar için ise LLaMA daha iyi özellikler sunmaktadır.

## GenAI ile grafik veritabanları birleştirildiğinde ne tür uygulamalar geliştirilebilir?

**Fraud Detection :** Dolandırıcılık Tespiti Sistemlerinde çok fazla anomali tespit edilmesi veri karmaşıklığına yol açabiliyor. Graph veri tabanları sayesinde anomalinin “neden” fraud olabileceği hakkında bilgi edinilebilir.

**Bioteknoloji Uygulamaları:** GenAI, protein, gen ve hastalıkların arasındaki ilişkilere dayanarak akılcı hipotezler geliştirebilir.

**Tavsiye Sistemleri:** Kullanıcılar, ürünler, kategoriler, geçmiş tıklamalar grafik veritabanında tutulur. GenAI bu ilişkilere dayalı olarak **kişiselleştirilmiş açıklamalar** üretir. Örnek, Hepsiburada.com.

**Chatbotlar**: Örneğin bir sağlık chatbotu, hastalık semptomları arasındaki ilişkileri grafik veri tabanında tutar. Chatbot hastayla konuşurken bağlamını graf veritabanından alır.

**İnsan Kaynakları Yetenek Eşleştirme Asistanı:** Kişiler, beceriler, deneyimler ,şirketler ilişkisel olarak graph veritabanında tutulur.Yüzlerce başvuru arasından bu bağlamda doğru adaylar seçilmeye çalışılır.

LLM’ler çok yaratıcı sonuçlar döndürse de bazen hatalı sonuçlar döndürebilir. LLM ve Graph birleştiğinde graph veri bağlantısı sayesinde daha yüksek doğruluk sağlanır. Sonuç olarak bu sistemlerde Graf veritabanı ilişkiyi, GenAI yorumu üretir.

## Riskleri veya etik tartışmaları var mıdır?

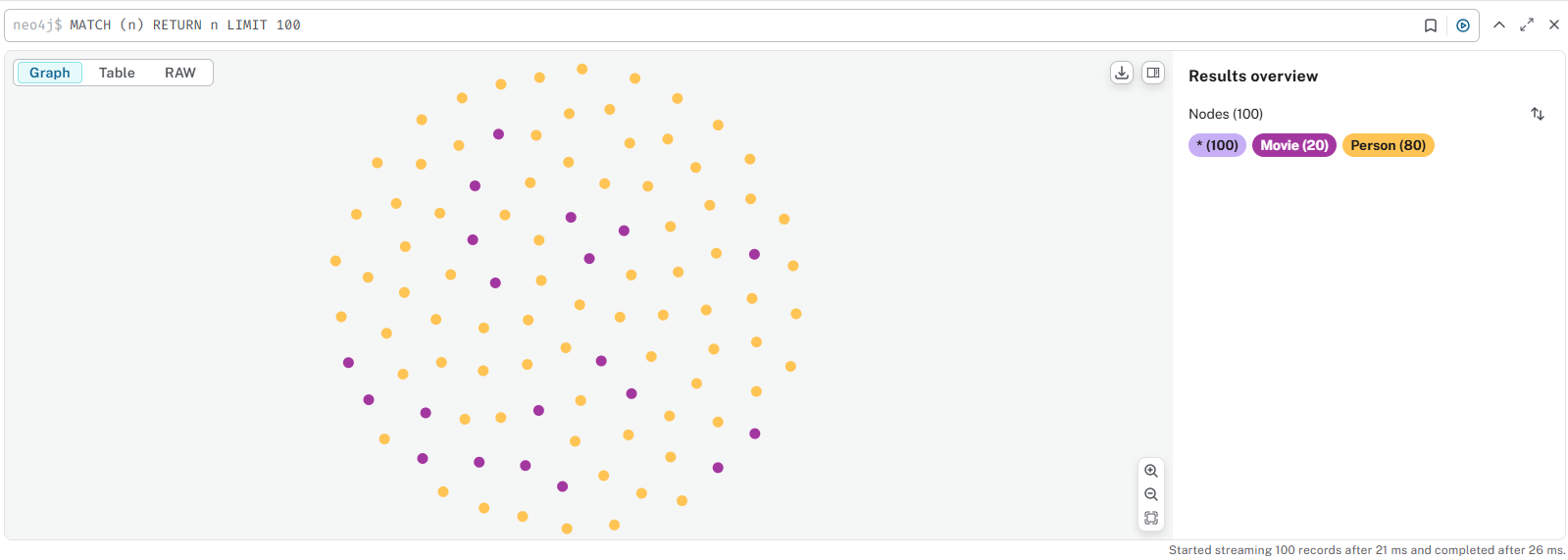
GenAI tarafından üretilen içeriğin sorumluluğunu kimin taşıdığı ve içeriklerin nasıl üretildiğine dair hesap verilebilirlik konusunda genel olarak endişeler var. Daha şimdiden bir sürü telif davası açılmış durumda. Model bire bir aynı içeriği üretmese bile esinlendiği eserlere göre kimin telif hakkı olduğu belirsiz bir duruma dönüştü. Ayrıca GenAI sistemleri gelişip daha özerk hale geldikçe, bu sistemlerin nasıl çalıştığına dair kontrolün kaybedilmesi de başka bir endişeye yol açabilir.

# Bölüm 2: Veri Yükleme ve Explore Arayüzü ile Çalışma

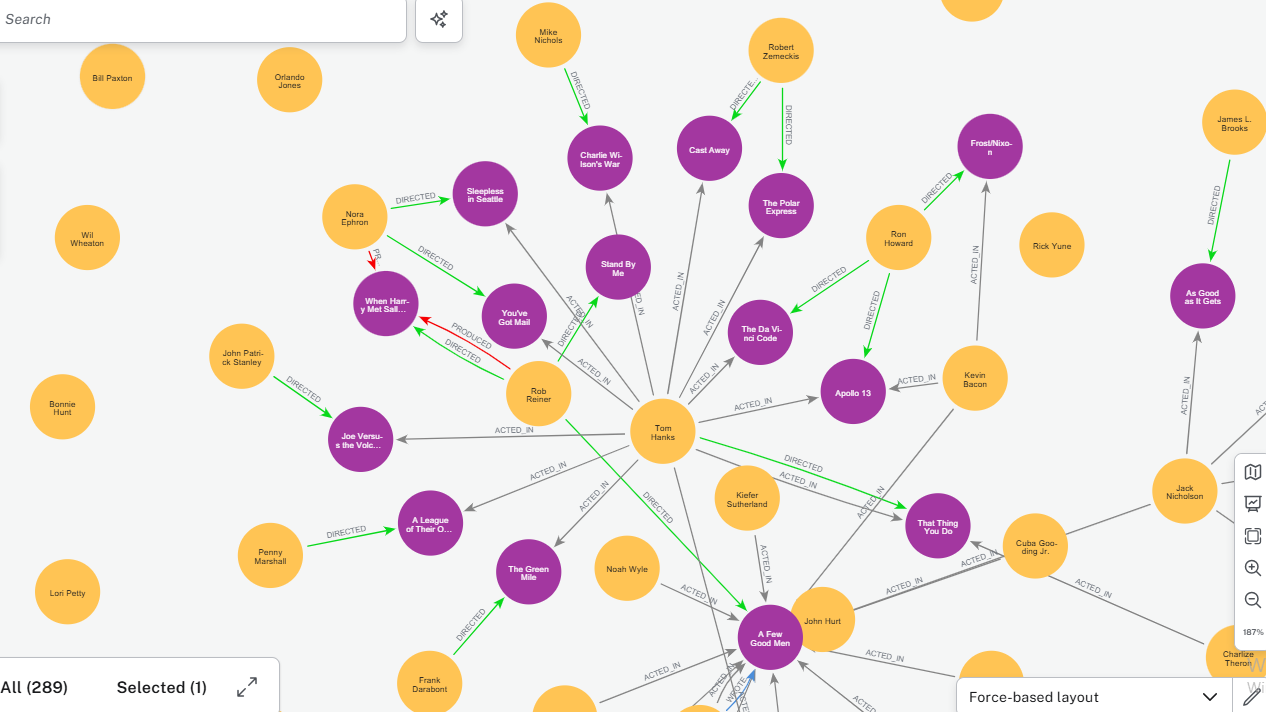
## Adım 1: Movies Veri Setini Yükleyin

NEO4J Desktop indirdim fakat Explore kısmı Cloud’daki gibi yoktu. Bundan dolayı sorguları desktoptan yapsam da Explore kısımları için Cloud’u kullandım.

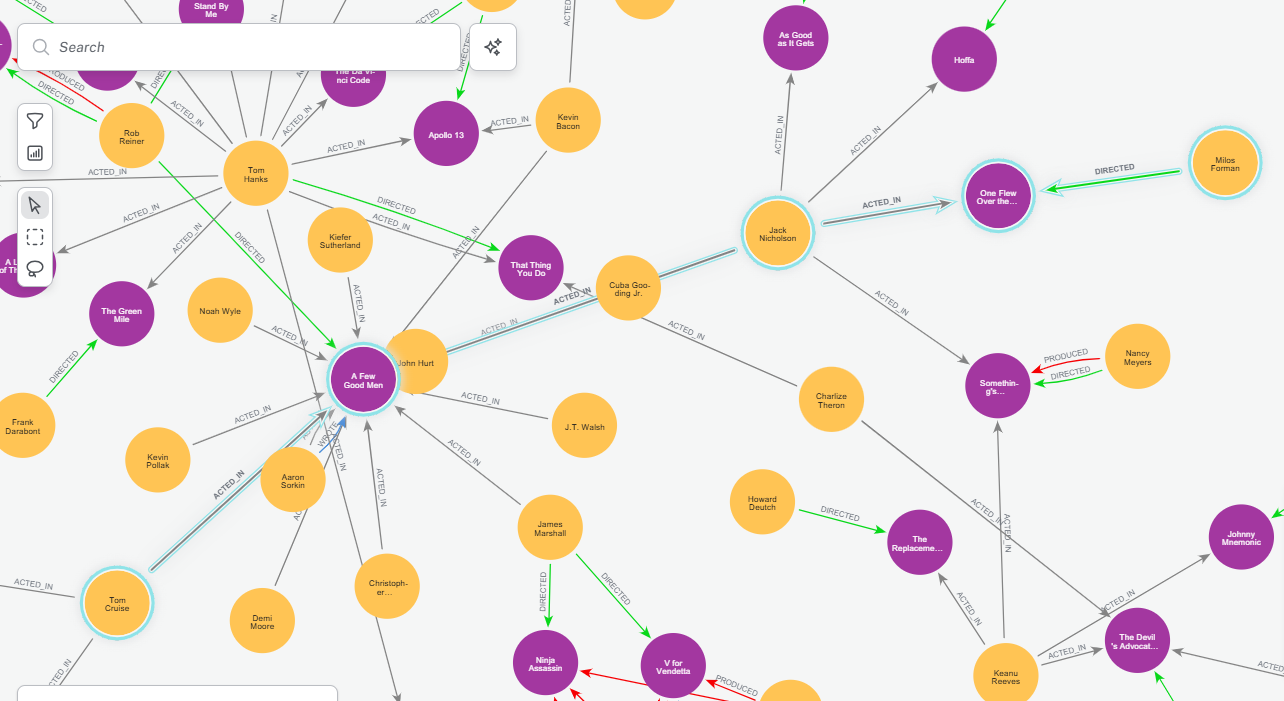
## Adım 2: Explore Sekmesinde Veriyi Keşfedin



MATCH (n) RETURN n LIMIT 100 sorgusunu çalıştırdığımda ekrandaki gibi 20 film ve 80 kişilik bir graf gördüm. Sarılar kişiyi, morlar filmi temsil ediyor.



Filmleri kimin yönettiğini görmek için relationship->DIRECTED bölümünü YEŞİL yaptım ve ekrandaki görüntüye ulaştım.

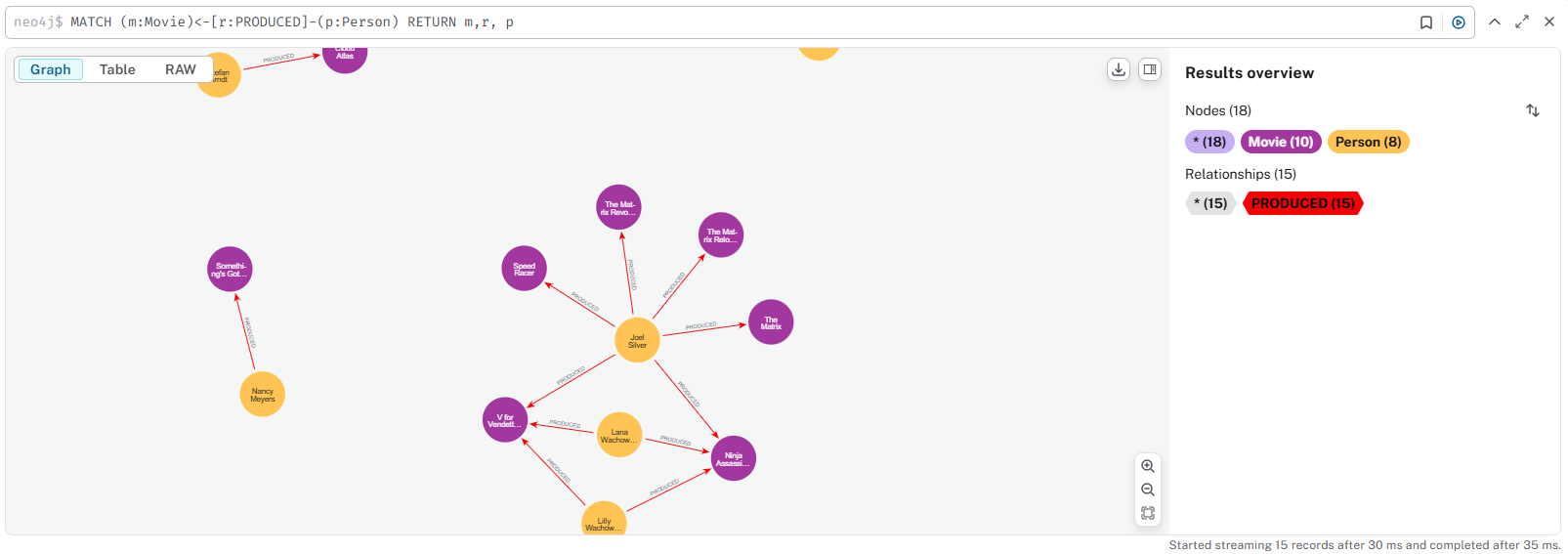


Shortest path fonksiyonu kullanarak. Tom Cruise’dan Milos Forman’a kaç adımda gidebileceğimi sorguladım. 5 adımda (2 film ve 3 kişi) ulaşabildiğimi gördüm.

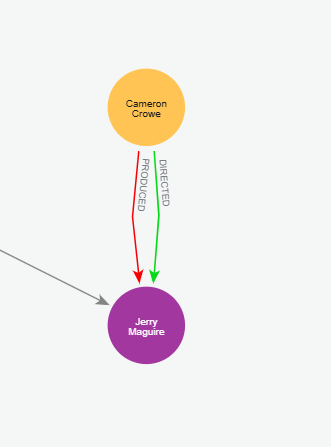
# Adım 3: Arama Alanıyla Özgün Sorgular Çalıştırın

# Görev:

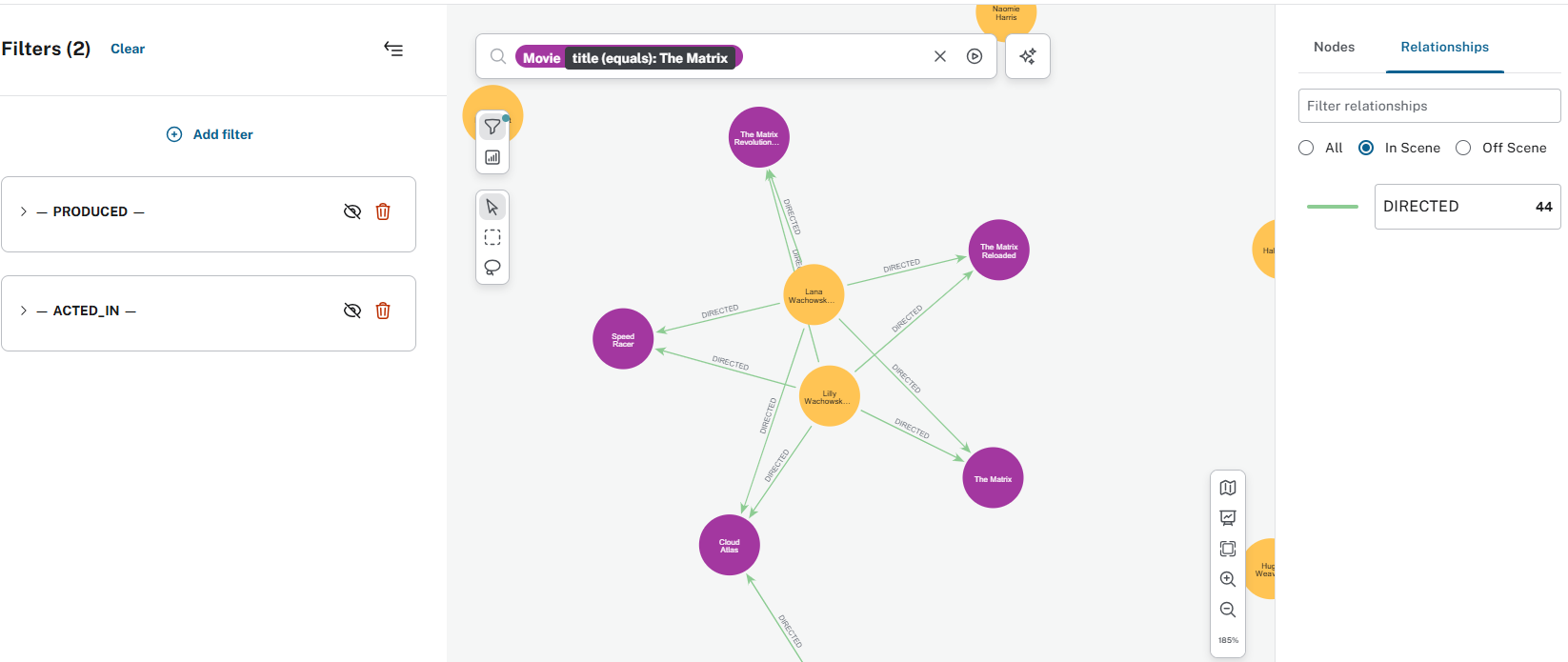
* En az 3 farklı arama yaparak farklı türde görseller elde edin.
* Bu aramaların ne ifade ettiğini 2-3 cümleyle açıklayın.
* Görsel olarak ilgç çıkan yapılarda hangi düğüm/ilişki çiftlerinin dikkat çekici olduğunu belirtin.



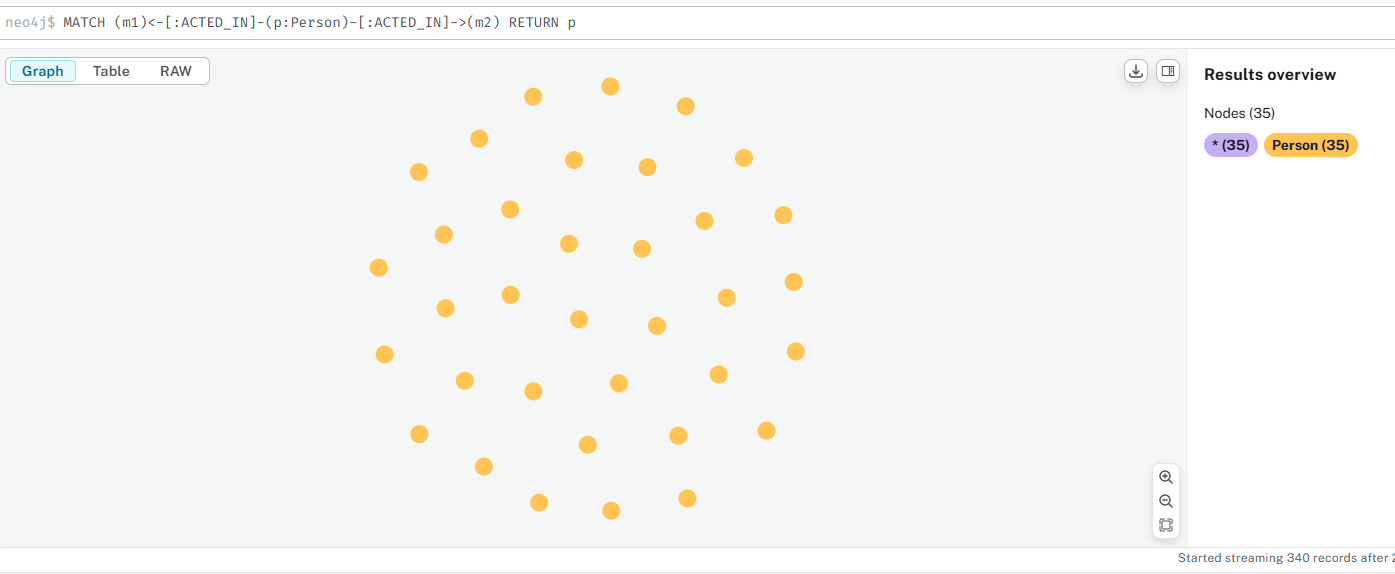
Film-Yapımcı-Üretim ilişkisini sorguladığımda sadece 8 yapımcı olduğunu ve en çok film üreten yapımcının “Joel Silver” olduğunu görüyorum.



Cameron Crowe, Jerry Maguire filminin hem yapımcılığını hem yönetmenliğini üstlenmiş.



Filtrelerden PRODUCED ve ACTED\_IN relationshiplerini gizledim. Ve film olarak “The Matrix”i seçtiğimde Lana ve Lily Wachowski kardeşlerin beraber 5 tane filmi birlikte yönettiklerini gördüm.



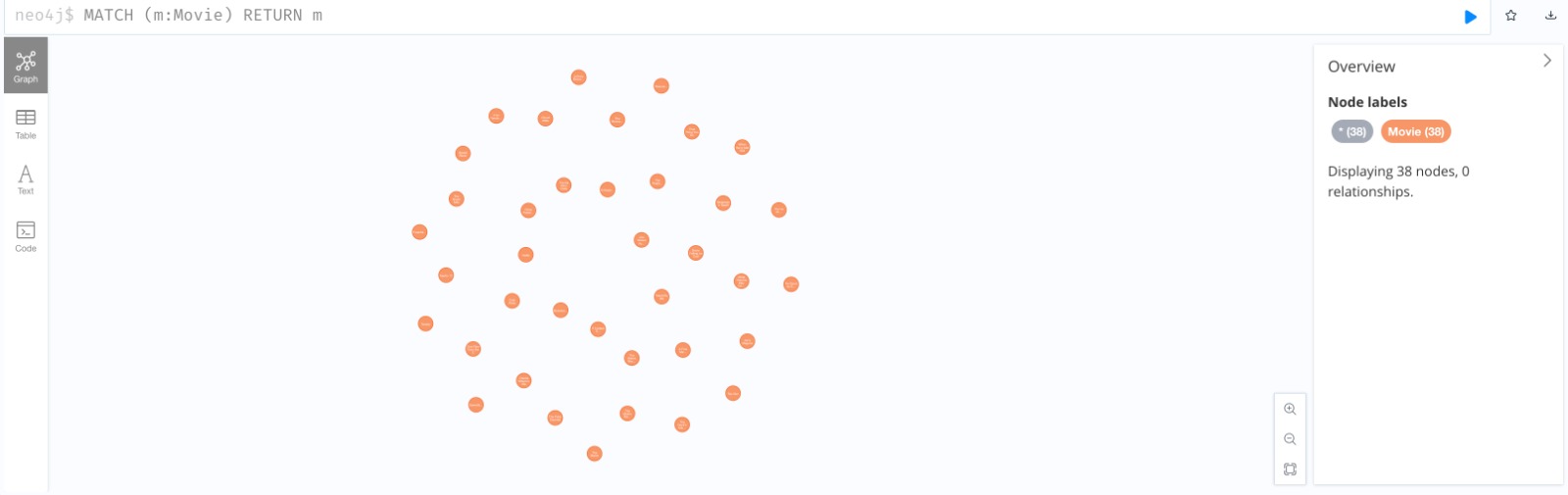
Birden fazla filmde oynayan 35 kişi olduğunu gördüm. Bunun için filmleri m1 ve m2 olarak gösterdim. Sadece sayıyı bulmak için count(p) de diyebilirdim.

# Bölüm 3: Cypher Sorguları ile Veriyi Keşfetme

Aşağıdaki soruları yanıtlarken Query sekmesine geçin ve her soruya uygun bir Cypher sorgusu yazın. Ardından sonucunu yorumlayarak raporlayın.

1. Veritabanındaki tüm film adlarını listeleyin.

MATCH (m:Movie) RETURN m

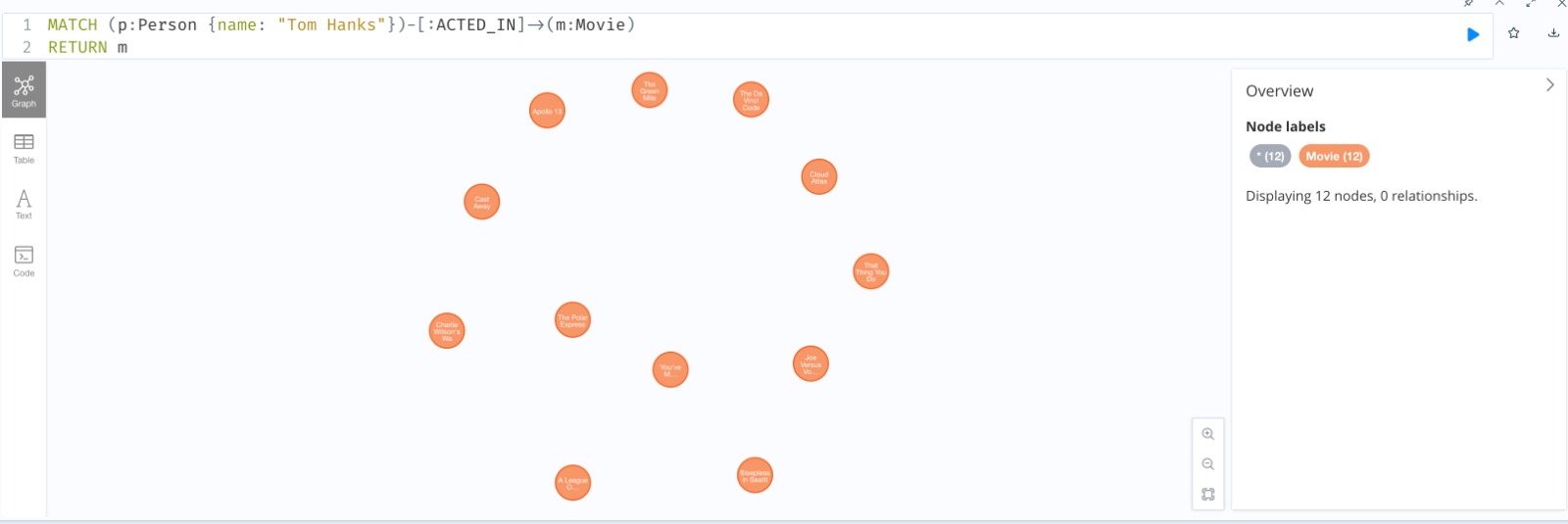


**Açıklama 1:** Sorgumuzda SQL dilinde SELECT işlevine sahip olan MATCH’I kullandım. Movie’leri “m” ataması olarak kullandım ve ekranda bu m nodelarını görmek için RETURN m komutunu kullandım. 38 tane film olduğunu gördüm.

1. “Tom Hanks”’in oynadığı filmleri bulun.

MATCH (p:Person {name: “Tom Hanks”}-[:ACTED\_IN]->(m:Movie)

RETURN m



**Açıklama 2:** Sorgumuzda name propertysi “Tom Hanks” olan Person’u “p”ye atadım. Filmleri m ile oynamasını ise ACTED\_IN relationshipi ile gösterdim. RETURN m diyerek Tom Hanks’ın oynadığı 12 film olduğunu 12 nodeda gördüm.

1. Her film için yönetmenlerini listeleyin.

MATCH (m:Movie)<-[:DIRECTED]-(director)

RETURN m.title as Movie ,director.name as Director



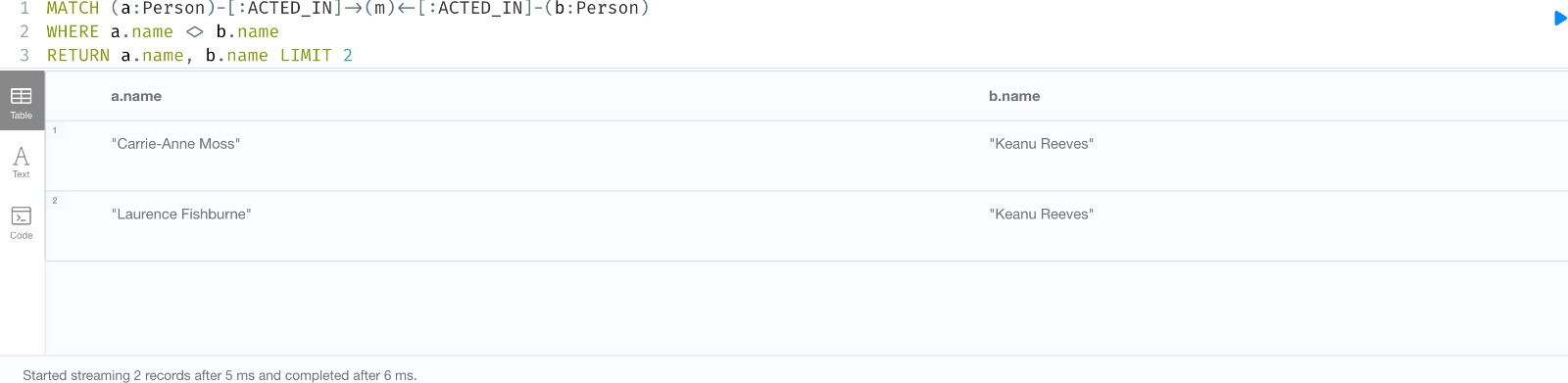
**Açıklama 3:** Bu sorguda öncelikle film-Yönetme-Yönetmen ilişkisini yazdım. Filmlere m, yönetmene director atamalarını yaptım. Ekranda film ve yönetmenini tablo şeklinde göstermek istediğim için RETURN ile filmin başlığını ve yönetmenin ismini getirdim.

1. Aynı filmde oynamış iki farklı oyuncu çiftini listeleyin (tekrarsız).

MATCH (a:Person)-[:ACTED\_IN]->(m)<-[:ACTED\_IN]-(b:Person)

WHERE a.name <> b.name

RETURN a.name, b. name LIMIT 2

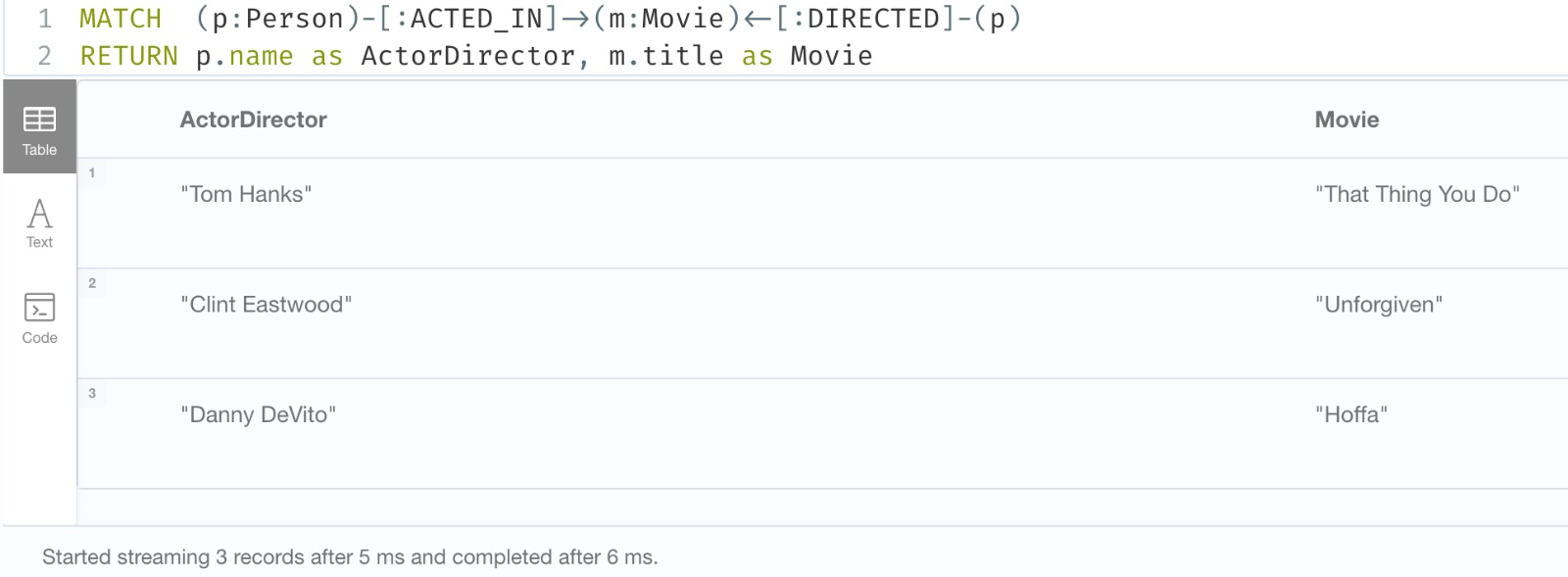


**Açıklama 4:** Bu sorguda ilk oyuncuyu **a**, ikincisini ise **b** olarak tanımladım. İkisinin de aynı filmde oynadığını göstermek için Filmleri merkez alan çift yönlü ilişki sorgusunu yazdım. Sorgumun tekrarsız olması için ise WHERE ile a ve b’nin isminin aynı olmaması gerektiği şartını koydum.

1. Hem yönetip hem oynadığı bir film olan kişileri ve filmleri bulun.

MATCH (p:Person)-[:ACTED\_IN]->(m:Movie)<-[:DIRECTED]-(p)

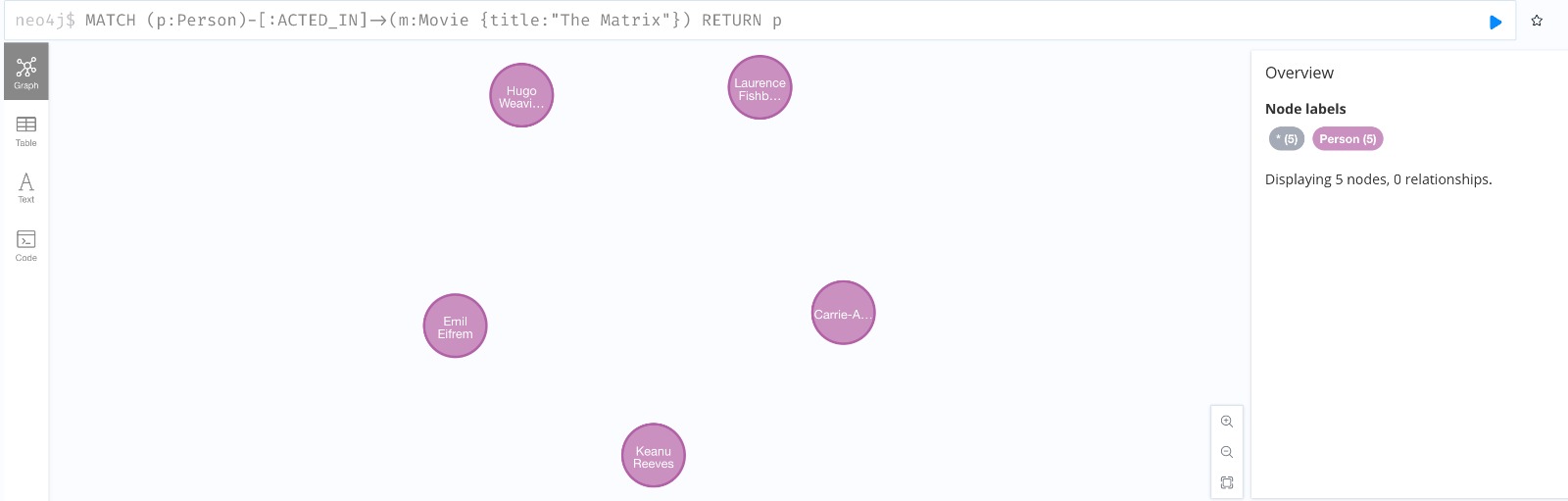
RETURN p.name as ActorDirector, m.title as Movie



**Açıklama 5:** Bu sorgu için filmi merkeze alan iki ilişkiyi birleştirdim. İlk ilişkiye ACTED\_IN, ikincisine DIRECTED koydum. Kişi için “p” ataması yapmıştım. O kişiyi getirmek için ilişkinin digger tarafına da “p” koydum. Tablo şeklinde düzgün bir gösterim almak için ise ALIAS’lar kullandım.

1. "The Matrix" filminde oynamış oyuncuları listeleyin.

MATCH (p:Person)-[:ACTED\_IN]->(m:Movie {title:"The Matrix"}) RETURN p



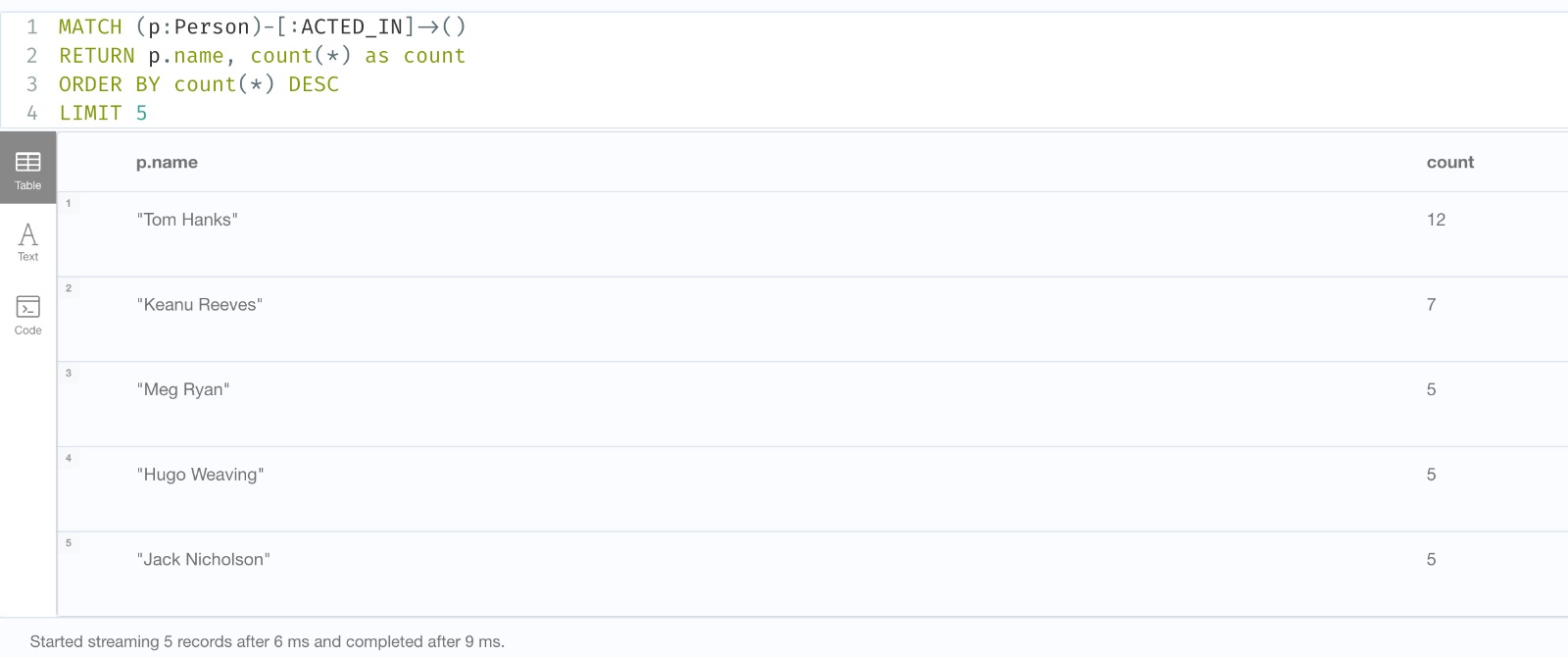
**Açıklama 6:** Kişileri “p” ile ilişkiyi ACTED\_IN ile gösterdikten sonra istediğim film için parantezin içine title: "The Matrix" propertysini koydum. Sadece kişileri getirmek için RETURN p ifadesini kullandım.

1. En çok filmde oynamış 5 kişiyi bulun.

MATCH (p:Person)-[:ACTED\_IN]->()

RETURN p.name , count(\*) as count

ORDER BY count(\*) DESC LIMIT 5

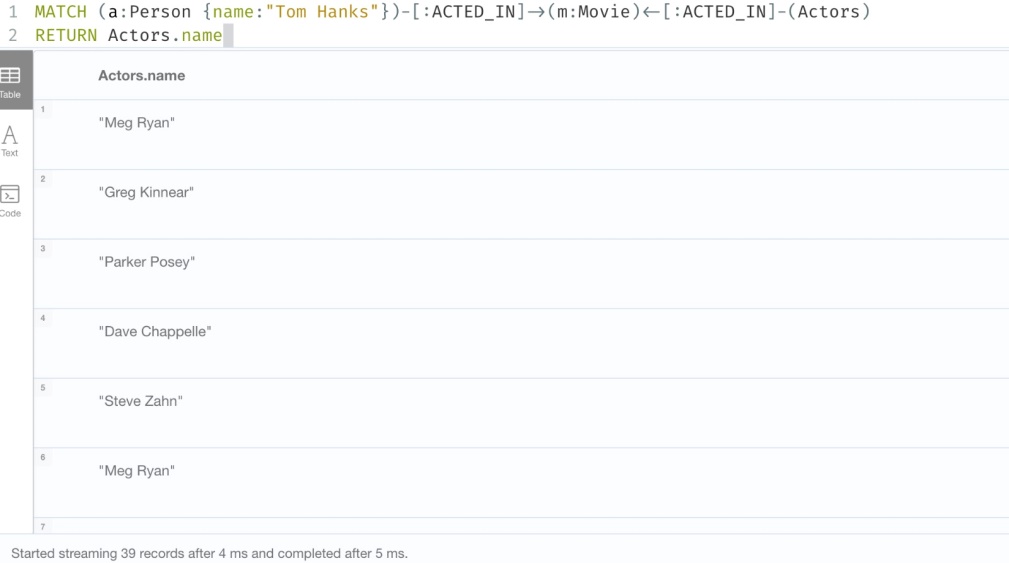


**Açıklama 7:** Bu sorguda 5 kişi görmek için LIMIT 5 ifadesini, büyükten küçüğe sıralamak için ORDER BY DESC komutunu, kaçar tane oynadıklarını görmek için ise COUNT aggregation fonksiyonunu kullandım. Kişi ve ilşkiyi yazdıktan sonra filmleri sorgumda getirmek istemediğim için boş parantez kullandım.

1. “Tom Hanks” ile aynı filmde oynamış diğer oyuncuları listeleyin.

MATCH (a:Person {name:"Tom Hanks"})-[:ACTED\_IN]->(m:Movie)<-[:ACTED\_IN]-(Actors)

RETURN Actors.name

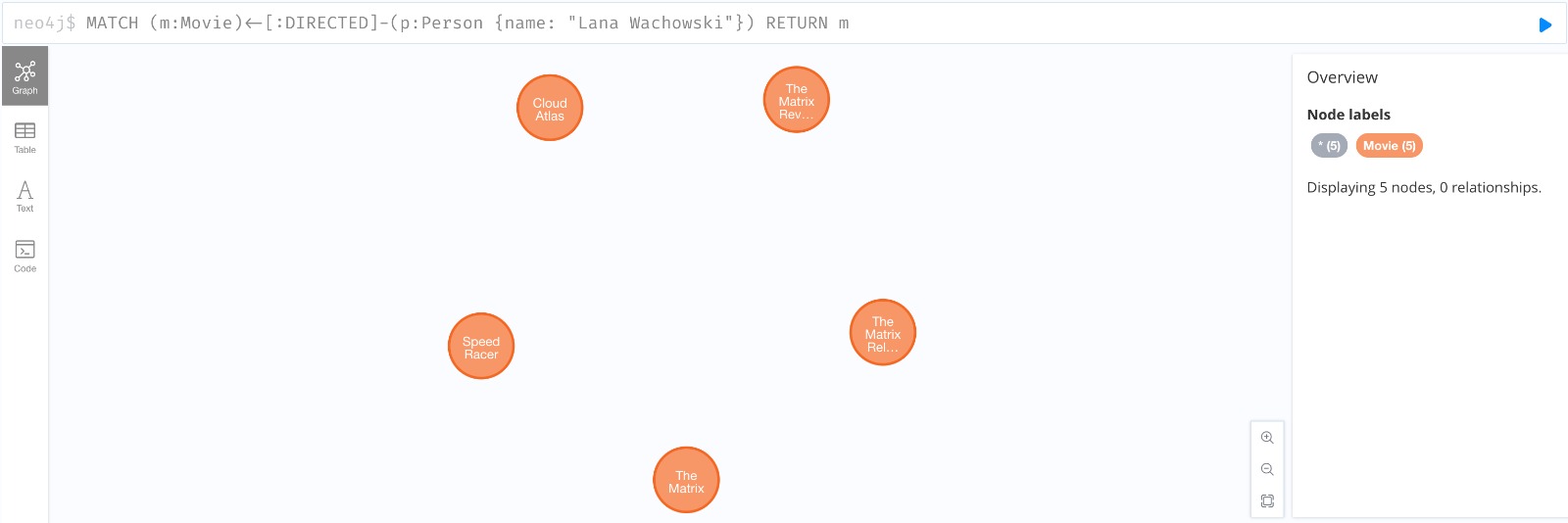


**Açıklama 8:** Bu sorguda yine ikili birleşik ilişki yapısını kullandım. İlk oyuncunun Tom Hanks olduğunu property olarak belirttim. Çift taraftan ACTED\_IN ile filmde oynadıklarını gösterdim. Ekranda diğer aktörleri görmek için ise Actors.name return’ü yaptım.

1. Yönetmeni "Lana Wachowski" olan filmleri bulun.

MATCH (m:Movie)<-[:DIRECTED]-(p:Person {name: "Lana Wachowski"})

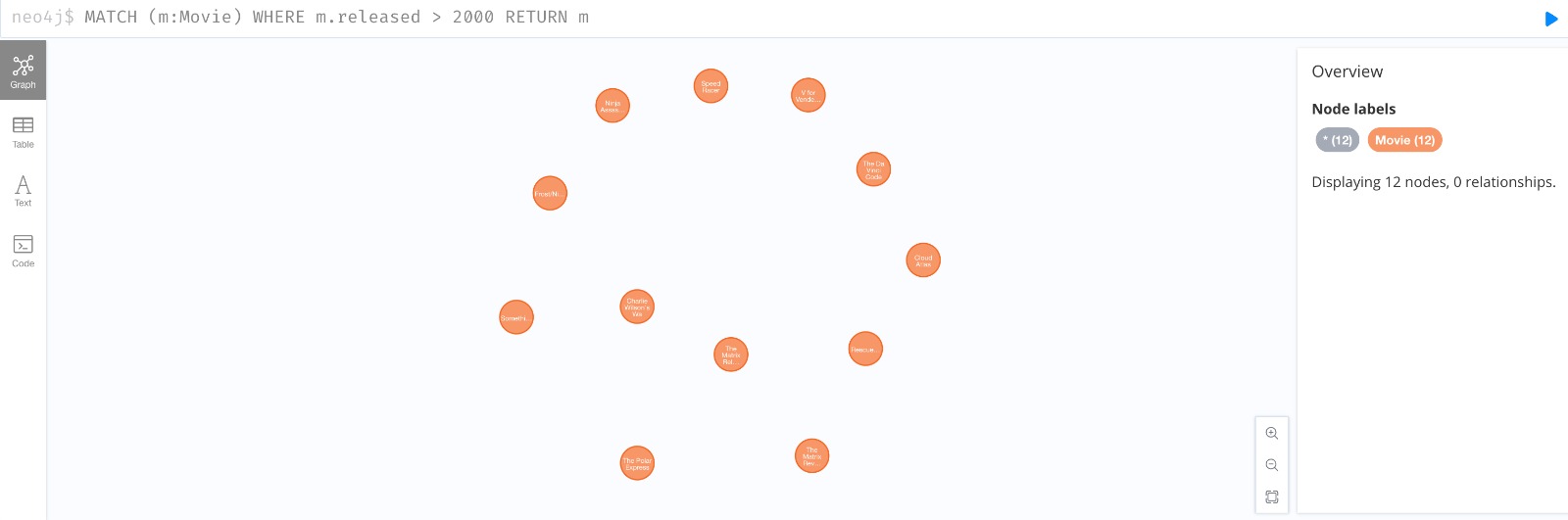
RETURN m



**Açıklama 9:** Yönetmen Lana Wachowski’yi person ve property kullanarak belirttim. DIRECTED ilişkisi ile yönettiği filmler bağlantısını kurdum. Ekrana ise m ataması ile filmleri getirdim.

1. 2000 yılından sonra yayınlanmış filmleri listeleyin.

MATCH (m:Movie) WHERE m.released > 2000 RETURN m



**Açıklama 10:** Ekranda görmek istediğim filmlere “m” ataması yaptım. İstenilen koşul için WHERE ve > komutunu kullandım. RETURN m ile bu koşula uyan filmleri getirdim.