

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

BUSENUR GÖKLER

### - Generative AI nedir?

İnsanların yaratıcılığını kopyalayarak geçmişten günümüze kadar varolan verileri işleyerek birtakım işlemler sonucunda kullanılarak yeni veriler(görsel, ses, kod, metin vs.) üretilebilen sistemlerdir.

### - Hangi alanlarda kullanılır? (en az 3 farklı alan)

İçerik üretimi, yazılım, pazarlama alanlarında kullanılabilir.

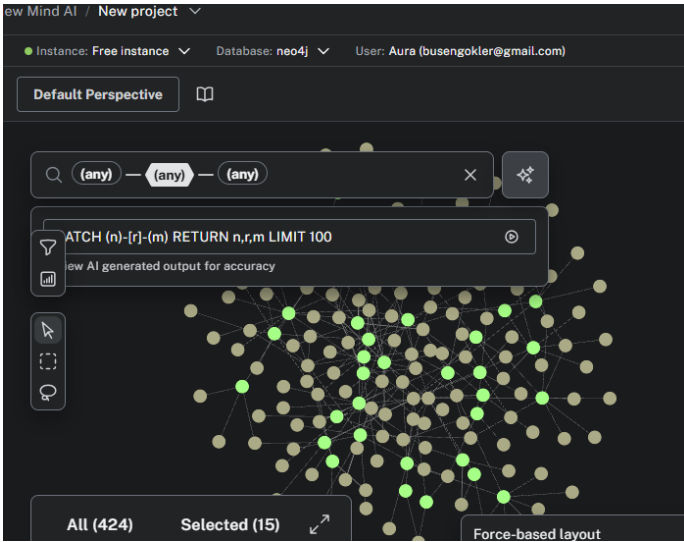
### - GenAI ile grafik veritabanları birleştirildiğinde ne tür uygulamalar geliştirilebilir?

Grafik veri tabanlarına normal veri tabanları gibi her türlü veriyi yükleyebildiğimiz için istediğimiz her alanda olabilecek verileri kullanarak cevap üretmeyi sağlayan uygulamalar olabilir. Örneğin, ortak meyve sebzeleri kullanan yemek tarifleri üretebilirsiniz.

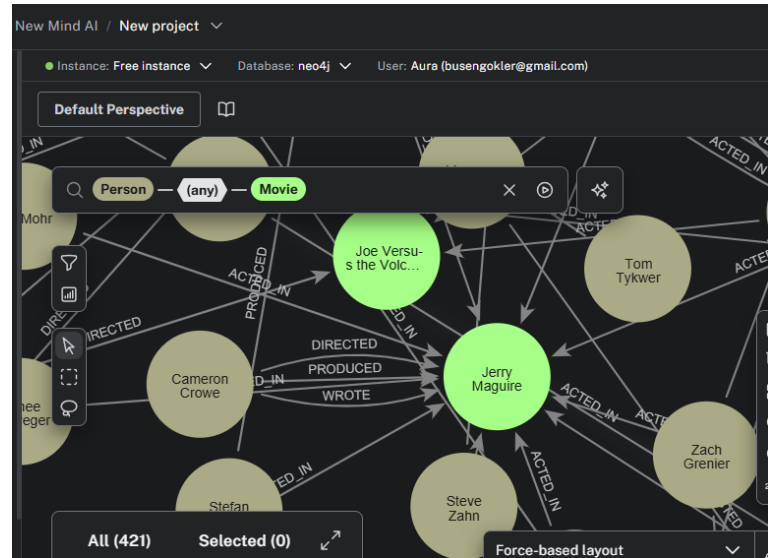
### - Riskleri veya etik tartışmaları var mıdır?

Verileri kullanan uygulamalarda her zaman erişim sebebiyle riskler olmaktadır, verilerin kimin elinde nasıl kullanıldığı bilinmemektedir. Bu verileri kullanarak sahteciliğe ve kötü kullanıma yol açabilir, burada da etik sorunlar ortaya çıkar.

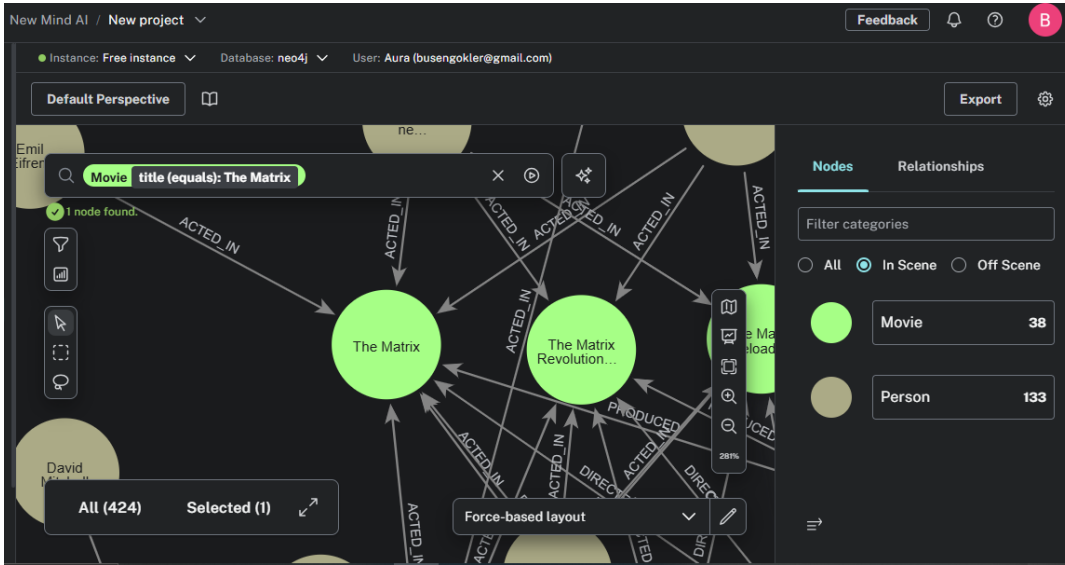
## Bölüm 2: Veri Yükleme ve Explore Arayüzü ile Çalışma



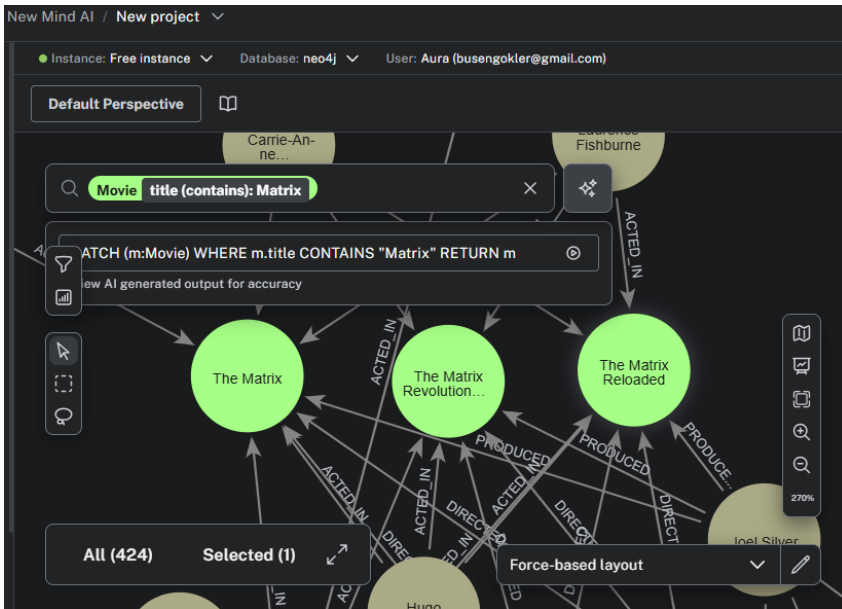
Burada tüm nodeları ve ilişkileri görüntüledim.



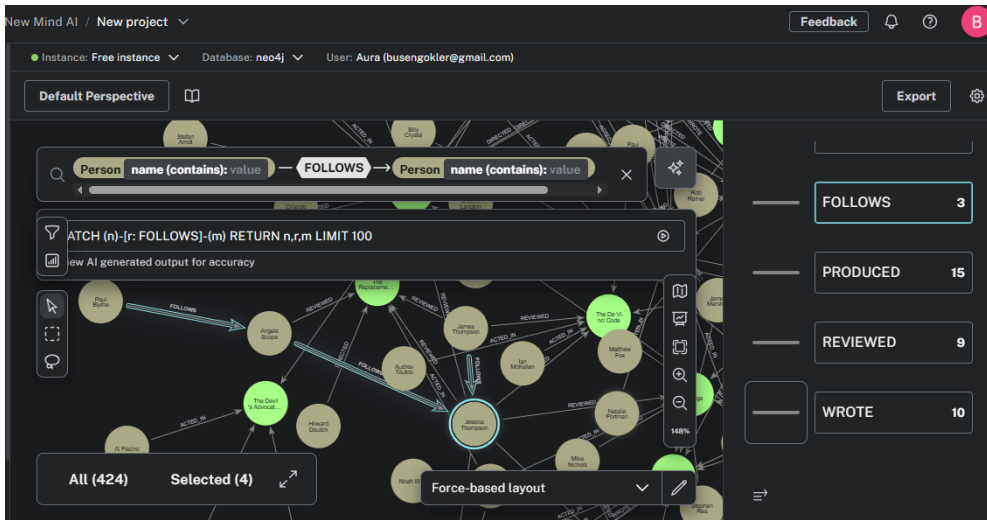
Cameron Crowe "Jerry Maguire" filmi hem yazmış hem yönetmiş hem de ürünleştirmiş olduğunu bir filmle birçok ilişkisinin bulunduğunu görmüş oldum.



Burada ismi(title)  
“The Matrix”  
olan filmi  
buldum.



Burada ise, içerisinde “Matrix”  
geçen filmleri listeledim.



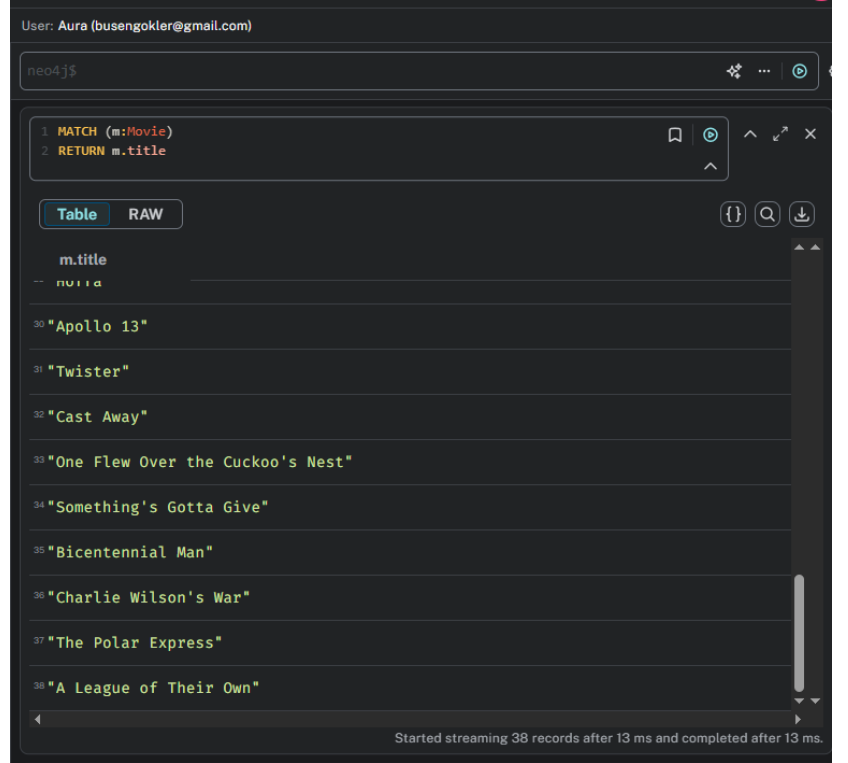
Burada, birbirini takip  
eden kişileri  
göüntüledim.

## Bölüm 3: Cypher Sorguları ile Veriyi Keşfetme

### 1. Veritabanındaki tüm film adlarını listeleyin.

**MATCH (m:Movie) RETURN m.title**

Sorgum, veritabanından "Movie" etiketi olan tüm nodeları bulur, Movie node'umu m adlı değişkene attım. Ardından m.title yazarak sadece filmlerin isimlerini görüntüledim. Bu sayede sonuçlar daha sade ve okunaklı hale geldi. Toplamda 38 adet film listelendi.



### 2. "Tom Hanks"'in oynadığı filmleri bulun.

**MATCH (p:Person {name:"Tom Hanks"})-[a:ACTED\_IN]-(m:Movie)**  
**RETURN m.title**

veya WHERE ile

**MATCH (p:Person)-[a:ACTED\_IN]-(m:Movie) WHERE p.name = "Tom Hanks" RETURN m.title**

Tom Hanks'in oynadığı filmler toplamda 12 tanedir. Filmlerin isimlerini görüntülemek için m.title yaptım. Sorguyu hem WHERE ile hem de node tanımlaması içerisinde nasıl yapabileceğimi yazdım.



### 3. Her film için yönetmenlerini listeleyin.

`MATCH (m:Movie)<-[d:DIRECTED]-(p:Person) RETURN p.name, m.title`

User: Aura (busengokler@gmail.com)

neo4j\$

neo4j\$ `MATCH (m:Movie)<-[d:DIRECTED]-(p:Person) RETURN p.name, m.title`

Table RAW

	p.name	m.title
1	"Lilly Wachowsk i"	"The Matrix"
2	"Lana Wachowsk i"	"The Matrix"
3	"Lilly Wachowsk i"	"The Matrix Rel oaded"
4	"Lana Wachowsk	"The Matrix Rel

Sorgulamamı yukarıdaki gibi yaptığımda filmin birden fazla yönetmeni varsa bile alt alta ayrı gözüküyordu ancak ben aynı filmin birden fazla yönetmeni varsa yan yana liste gibi gözüksün istiyordum.

Araştırdığımda COLLECT ile bunu yapabildiğimi gördüm. Bu şekilde daha iyi bir görüntü oluşmuş oldu.

Sonuçta 38 film için yönetmenleri listelenmiş oldu.

`MATCH (m:Movie)<-[d:DIRECTED]-(p:Person)`

`RETURN m.title as movie, COLLECT(p.name) AS directors`

User: Aura (busengokler@gmail.com)

neo4j\$

`1 MATCH (m:Movie)<-[d:DIRECTED]-(p:Person)`  
`2 RETURN m.title as movie, COLLECT(p.name) AS directors`

Table RAW

	movie	directors
1	"The Matrix"	["Lilly Wachowski", "Lana Wachowski"]
2	"The Matrix Reloaded"	["Lilly Wachowski", "Lana Wachowski"]
3	"The Matrix Revolutions"	["Lilly Wachowski", "Lana Wachowski"]
4	"The Devil's Advocate"	["Taylor Hackford"]
5	"A Few Good Men"	["Rob Reiner"]
6	"Top Gun"	["Tony Scott"]
7	"Jerry Maguire"	["Cameron Crowe"]

Started streaming 38 records after 30 ms and completed after 32 ms.

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

`MATCH (m:Movie)-[:ACTED_IN]-(p1:Person)`

`MATCH (m)-[:ACTED_IN]-(p2:Person)`

`WHERE p1.name < p2.name`

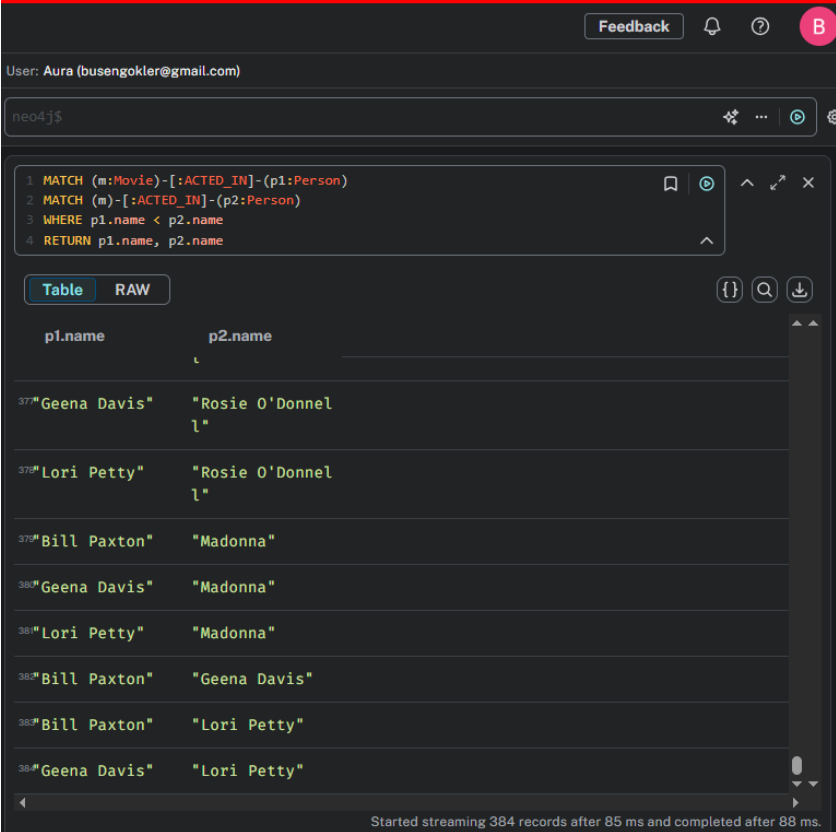
`RETURN p1.name, p2.name`

veya

```
MATCH (p2:Person)-[:ACTED_IN]->(m:Movie)-[:ACTED_IN]-(p1:Person)
```

```
WHERE p1.name < p2.name
```

```
RETURN p1.name, p2.name
```



The screenshot shows a Neo4j Cypher query interface. The query is:

```
1 MATCH (m:Movie)-[:ACTED_IN]-(p1:Person)
2 MATCH (m)-[:ACTED_IN]-(p2:Person)
3 WHERE p1.name < p2.name
4 RETURN p1.name, p2.name
```

The results are displayed in a table view with two columns: p1.name and p2.name. The table contains 8 rows of data, each representing a pair of actors who have appeared in the same movie. The results are:

p1.name	p2.name
37# "Geena Davis"	"Rosie O'Donnel l"
37# "Lori Petty"	"Rosie O'Donnel l"
37# "Bill Paxton"	"Madonna"
38# "Geena Davis"	"Madonna"
38# "Lori Petty"	"Madonna"
38# "Bill Paxton"	"Geena Davis"
38# "Bill Paxton"	"Lori Petty"
38# "Geena Davis"	"Lori Petty"

At the bottom of the interface, it says: "Started streaming 384 records after 85 ms and completed after 88 ms."

İlk sorgumda, "Movie" node ile bağlantılı olan "Person" nodelarını eşleştirdim. p1 ve p2 adında iki farklı oyuncuyu temsil eden değişkenler kullandım. WHERE p1.name < p2.name şartıyla aynı eşleşmenin tersini (yani tekrarını) engelledim. Sonuç olarak birlikte oynayan oyuncu ikililerini, isimleriyle birlikte listelemiş oldum.

İkinci sorgumda ise, Person ve Movie nodeları arasındaki ACTED\_IN ilişkilerini yönlü şekilde kullandım. p2 isimli kişinin bir filmde oynadığını, aynı filme p1 isimli başka bir kişinin de dahil olduğunu belirttim.

İki sorgu da aynı mantıkla çalışıyor ama bu versiyonda oklar sayesinde ilişkilerin yönü daha açık bir şekilde belirtilmiş oluyor. Sorgumu çalıştırdığımda ise 384 sonuç çıktı, tekrarlı haline göre neredeyse yarıya düştü.

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

**MATCH (p:Person)-[d:DIRECTED]-(m:Movie)-[a:ACTED\_IN]-(p)**

**RETURN p.name,m.title**

Bu sorguda, 3 kişi ve film listelendi. Sorguyu aynı zamanda Explore kısmında da denedim ve inceledim. Ancak Explore kısmında doğru sonuçları vermemiştii nedeninin AI'ın Query'i anlamakta çok iyi çalışmaması olduğunu düşünüyorum, ilişki yönünü yanlış yapmıştı.

User: Aura (busengokler@gmail.com)

neo4j\$

```
1 MATCH (p:Person)-[d:DIRECTED]-(m:Movie)-[a:ACTED_IN]-(p)
2 RETURN p.name,m.title
```

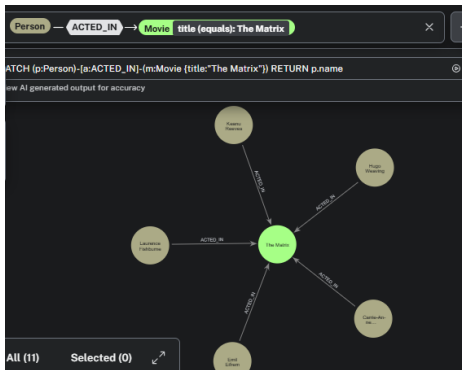
	p.name	m.title
1	"Tom Hanks"	"That Thing You Do"
2	"Clint Eastwood"	"Unforgiven"
3	"Danny DeVito"	"Hoffa"

Started streaming 3 records after 49 ms and completed after 51 ms.

## 6. "The Matrix" filminde oynamış oyuncuları listeleyin.

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

Filtrelemeyi(title) node içerisinde yaptım, sorgum sonucunda 5 oyuncu listelendi, bu sorguyu projenin başında Explore kısmında da denemiştii ve incelemiştii.



User: Aura (busengokler@gmail.com)

neo4j\$

```
neo4j$ MATCH (p:Person)-[a:ACTED_IN]-(m:Movie {title:"The Matrix"}) RETURN p.name
```

	p.name
1	"Keanu Reeves"
2	"Carrie-Anne Moss"
3	"Laurence Fishburne"
4	"Hugo Weaving"
5	"Emil Eifrem"

Started streaming 5 records after 57 ms and completed

## 7. En çok filmde oynamış 5 kişiyi bulun.

**MATCH (p:Person)-[a:ACTED\_IN]-(m:Movie)**

**RETURN p.name, COUNT(m) as mc**

**ORDER BY mc DESC**

**LIMIT 5**

Oyuncuların oynadığı filmleri COUNT metodu ile buldum ve mc (movie count) diye adlandırdım. ORDER BY'ı mc filtresine göre yaptım ve sıralama büyükten küçüğe olması için DESC kullandım. LIMIT ile de sadece 5 sonuç listeledim. Sorgum sonucunda beklendiği gibi en çok filmde oynayan oyuncunun Tom Hanks olduğunu görüntüledim.

User: Aura (busengokler@gmail.com)

neo4j\$

```
1 MATCH (p:Person)-[a:ACTED_IN]-(m:Movie)
2 RETURN p.name, COUNT(m) as mc
3 ORDER BY mc DESC
4 LIMIT 5
```

Table RAW

	p.name	mc
1	"Tom Hanks"	12
2	"Keanu Reeves"	7
3	"Meg Ryan"	5
4	"Hugo Weaving"	5
5	"Jack Nicholson"	5

Started streaming 5 records after 41 ms

## 8. "Tom Hanks" ile aynı filmde oynamış diğer oyuncuları listeleyin.

**MATCH (p:Person {name:"Tom Hanks"})-[a:ACTED\_IN]-(m:Movie)-[:ACTED\_IN]-(p2:Person)**

**RETURN p2.name**

Tom Hanks'in oynadığı node'a p dedim, diğer oyuncular için ise farklı olabilmesi için p2 olarak adlandırdım. İki tarafta aynı filmde oynayacağı için aynı node'a(Movie) bağladım. Tom Hanks ile aynı filmde oynayan 39 oyuncu olduğunu görüntüledim.

User: Aura (busengokler@gmail.com)

neo4j\$

```
1 MATCH (p:Person {name:"Tom Hanks"})-[a:ACTED_IN]-(m:Movie)-[:ACTED_IN]-(p2:Person)
2 RETURN p2.name
```

Table RAW

	p2.name
31	"Bill Paxton"
32	"Helen Hunt"
33	"Philip Seymour Hoffman"
34	"Julia Roberts"
35	"Rosie O'Donnell"
36	"Bill Paxton"
37	"Madonna"

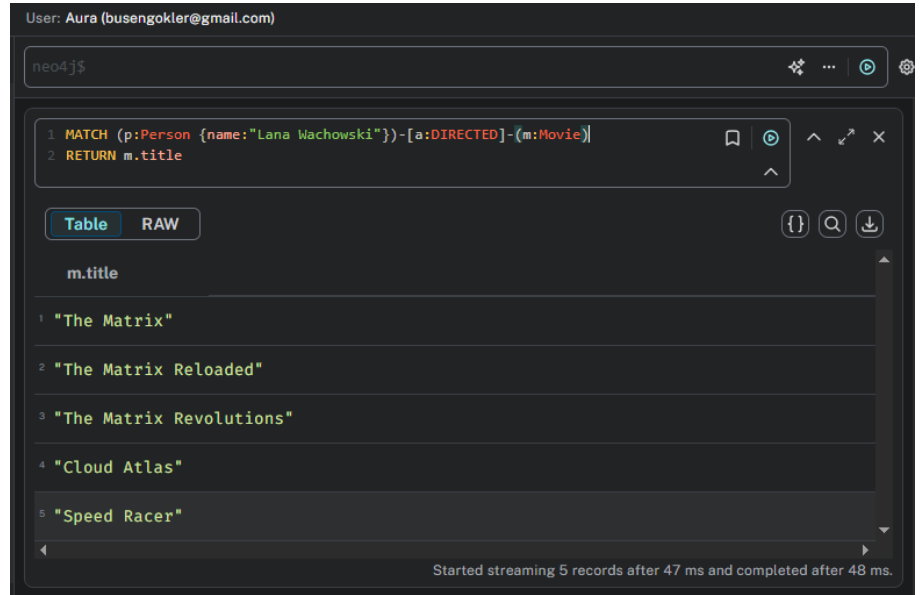
## 9. Yönetmeni "Lana Wachowski" olan filmleri bulun.

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

**RETURN m.title**

Sorgumda "Lana Wachowski"nin yönettiği filmlerin isimlerini yazdırdım.

Sorgu sonucunda Lana Wachowski'nin yönettiği 5 film olduğunu görüntüledim. Filmlerin zaten çok bilinen filmler olduğunu, birçok kişinin favorisi olan filmleri yönettiğini görmüş oldum.



## 10. 2000 yılından sonra yayınlanmış filmleri listeleyin.

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

**RETURN m.title**

2000 yılından sonra yayınlanan filmlerin sadece isimlerini görüntüledim ve toplamda 12 film olduğu sonucu aldım. Burada Explore kısmında herhangi bir Movie node'unu inceleyerek yayın tarihini "released" attribute'u ile tuttuklarını gördüm, sorgumu buna göre oluşturdum.

