**Cypher Query Language**

**Nedir**

Cypher, Neo4j'in etkileyici ve etkili veri sorgulamaya olanak tanıyan [bildirimsel](https://en-m-wikipedia-org.translate.goog/wiki/Declarative_programming?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) bir grafik sorgulama dilidir. 2011 yılında Neo4j mühendisleri tarafından grafik veritabanları için SQL'e eşdeğer bir dil olarak yaratılmıştır. SQL'e benzer şekilde Cypher, kullanıcıların grafikten ne alacaklarına odaklanmalarını sağlar*,* Bu nedenle Cypher, kullanıcıların daha önce bilinmeyen veri bağlantılarını ve kümelerini ortaya çıkaran verimli ve açıklayıcı sorgulara izin vererek özellik grafik veritabanlarının tüm potansiyelini gerçekleştirmelerini sağlar. Diğer dillere benzerliği ve sezgiselliği nedeniyle öğrenmesi açık ara en kolay grafik dilidir.

A blue dots on a black background

Description automatically generated

[Cypher, verileri düğümlere ve kenarlara (Cypher'da "ilişkiler" olarak adlandırılır) düzenleyen Property Graph Model'e](https://en-m-wikipedia-org.translate.goog/wiki/Graph_database?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc#Labeled-property_graph) dayanır . Düğümlerin ve ilişkilerin bu standart [grafik](https://en-m-wikipedia-org.translate.goog/wiki/Graph_(discrete_mathematics)?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) öğelerine ek olarak , property graph modeli, verilerin daha ince kategorilerini ve niteliklerini tanımlamak için etiketler ve özellikler ekler.

**Çalışma Prensibi**

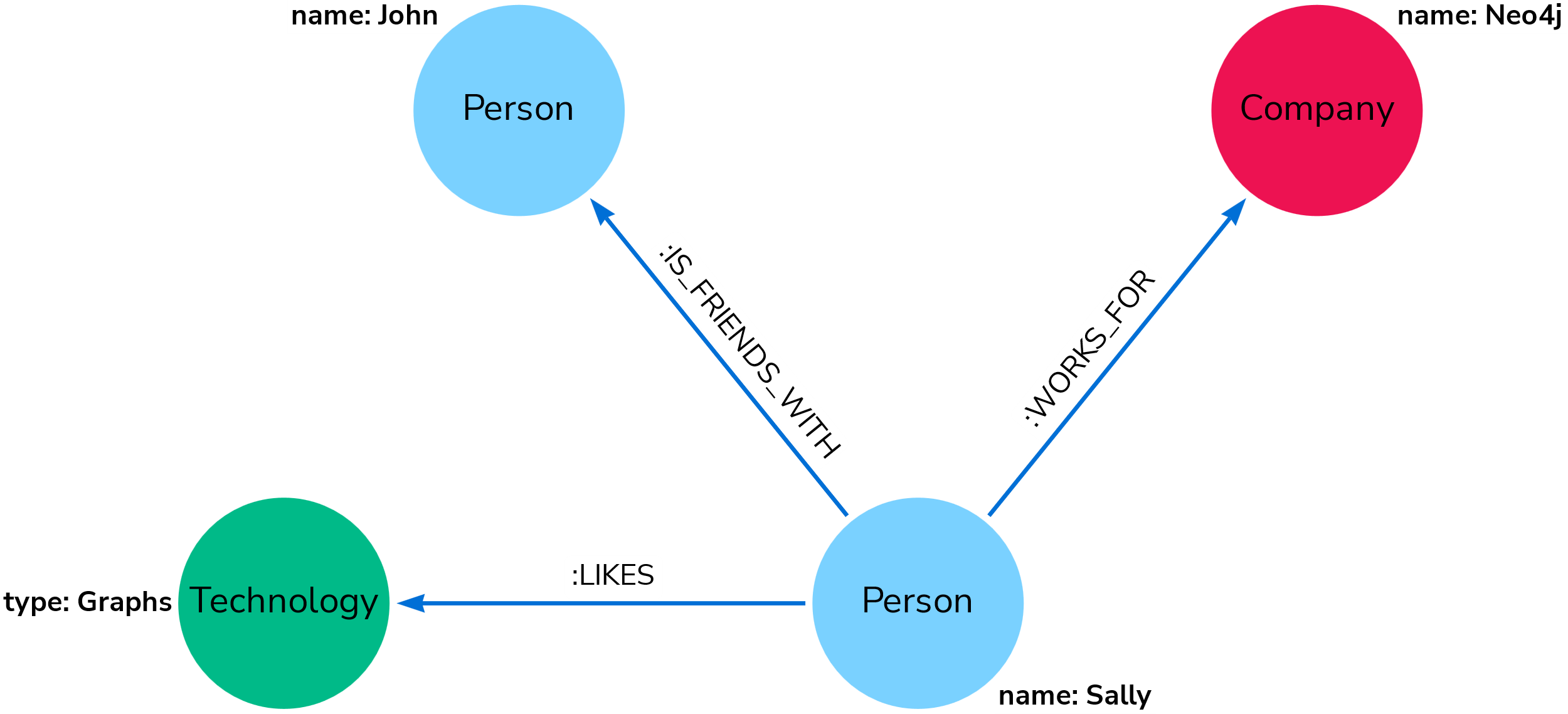
Desen tanıma, beynin çalışma biçiminin temelini oluşturur. Bu nedenle, insanlar desenlerle çalışmada çok iyidir (görsel diyagramları veya hatta hafıza eşleştirme oyunlarını düşünün). Cypher da büyük ölçüde desenlere dayanır ve verilerdeki bu desenlerin çeşitli versiyonlarını tanımak üzere tasarlanmıştır, bu da onu kullanıcıların öğrenmesi için basit ve mantıklı bir dil haline getirir.

Düğümler, grafikteki varlıklardır. Bunlar, özellikler adı verilen herhangi bir sayıda öznitelik (Key-Value çifti) tutabilir. Düğümler, bir etki alanındaki farklı rollerini temsil eden sıfır veya daha fazla etiketle (etiketler veya kategoriler gibi) etiketlenebilir. İlişkiler, iki düğüm varlığı arasında yönlendirilmiş, adlandırılmış, anlamsal olarak ilgili bağlantılar sağlar. Bir ilişkinin her zaman bir yönü, bir başlangıç ​​düğümü, bir bitiş düğümü ve tam olarak bir ilişki türü vardır. Düğümler gibi, ilişkilerin de özellikleri olabilir.



Etiketler, sıfır veya daha fazla düğüm etiketi atayarak benzer düğümleri bir araya toplayabilir. Etiketler bir nevi etiketlere benzer ve aranacak veya oluşturulacak belirli varlık türlerini belirtmenize olanak tanır. Özellikler, bir dize anahtarının ve Cypher tür sisteminden bazı değerlerin bağlanmasıyla anahtar-değer çiftleridir. Cypher sorguları, belirtilen desende bulunan verileri oluşturmak, okumak, güncellemek ve silmek için etiketler ve özellikler üzerinde belirtilen herhangi bir filtrelemeyle düğüm ve ilişki desenleriyle birleştirilir.

Cypher sorgu dili, düğüm ve ilişki desenlerini tasvir eder ve bu desenleri etiketlere ve özelliklere göre filtreler. Cypher'ın sözdizimi, bilgisayarlar için metin tabanlı görsel sanat olan [ASCII sanatına](https://en-m-wikipedia-org.translate.goog/wiki/ASCII_art?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) dayanır . Bu, dili çok görsel ve okunması kolay hale getirir çünkü sorguda belirtilen verileri hem görsel hem de yapısal olarak temsil eder. Örneğin, düğümler, öznitelikler ve varlıkla ilgili bilgiler etrafında parantezlerle temsil edilir. İlişkiler, ilişki türünün parantez içinde olduğu bir okla (yönlendirilmiş veya yönlendirilmemiş) tasvir edilir.

****

Cypher tür sistemi, diğer programlama ve sorgu dillerinde kullanılan yaygın türlerin çoğunu içerir. Desteklenen türler arasında boolean, string, number, integer ve floating-point sayıları gibi skaler değer türleri bulunur. Ayrıca , datetime, localdatetimedate, time, localtime ve duration gibi zamansal türleri de destekler. Haritalar ve listeler için kapsayıcı türleri, düğüm, ilişki ve yol için grafik türleri ve bir boşluk türü mevcuttur.

Diğer sorgu dillerine benzer şekilde, Cypher desenleri belirtmek, desenleri filtrelemek ve sonuçları döndürmek için çeşitli anahtar sözcükler içerir. En yaygın olanlar şunlardır: MATCH, WHERE ve RETURN. Bunlar SQL'deki SELECT ve WHERE'den biraz farklı çalışır ancak benzer amaçları vardır.

**Cypher ve SQL: temel farklar**

Cypher dili, [SQL'in](https://en-m-wikipedia-org.translate.goog/wiki/SQL?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc)gücü ve yeteneği düşünülerek tasarlanmıştır, ancak Cypher, [grafik teorisi](https://en-m-wikipedia-org.translate.goog/wiki/Graph_theory?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) kavramları üzerine inşa edilmiş bir veritabanının bileşenlerine ve ihtiyaçlarına dayanmaktadır . Bir grafik modelinde, veriler düğümler (matematik ve ağ biliminde [köşeler](https://en-m-wikipedia-org.translate.goog/wiki/Vertex_(graph_theory)?_x_tr_sl=en&_x_tr_tl=tr&_x_tr_hl=tr&_x_tr_pto=tc) ) ve ilişkiler (matematik ve ağ biliminde kenarlar) olarak yapılandırılır ve verilerdeki varlıkların birbirleriyle nasıl bağlandığına ve ilişkilendirildiğine odaklanılır.

Bir sorgunun örnek kullanımını görebilmek açısından hem SQL hem de Cypher karşılığına bakalım:

Alttaki sorgu her ikisorgu en pahallı 10 ürün ismi ve fiyatını gösterir.

**SQL:**

SELECT p.ProductName, p.UnitPrice

FROM products as p

ORDER BY p.UnitPrice DESC

LIMIT 10;

**Cypher:**

MATCH (p:Product)

RETURN p.productName, p.unitPrice

ORDER BY p.unitPrice DESC

LIMIT 10;

Şimdi de Cypher’ın SQL den daha özlü bir sorgu dii olduğunu alttaki aynı sonuçları veren sorgular ile inceleyelim.

**SQL:**

SELECT actors.name

FROM actors

LEFT JOIN acted\_in ON acted\_in.actor\_id = actors.id

LEFT JOIN movies ON movies.id = acted\_in.movie\_id

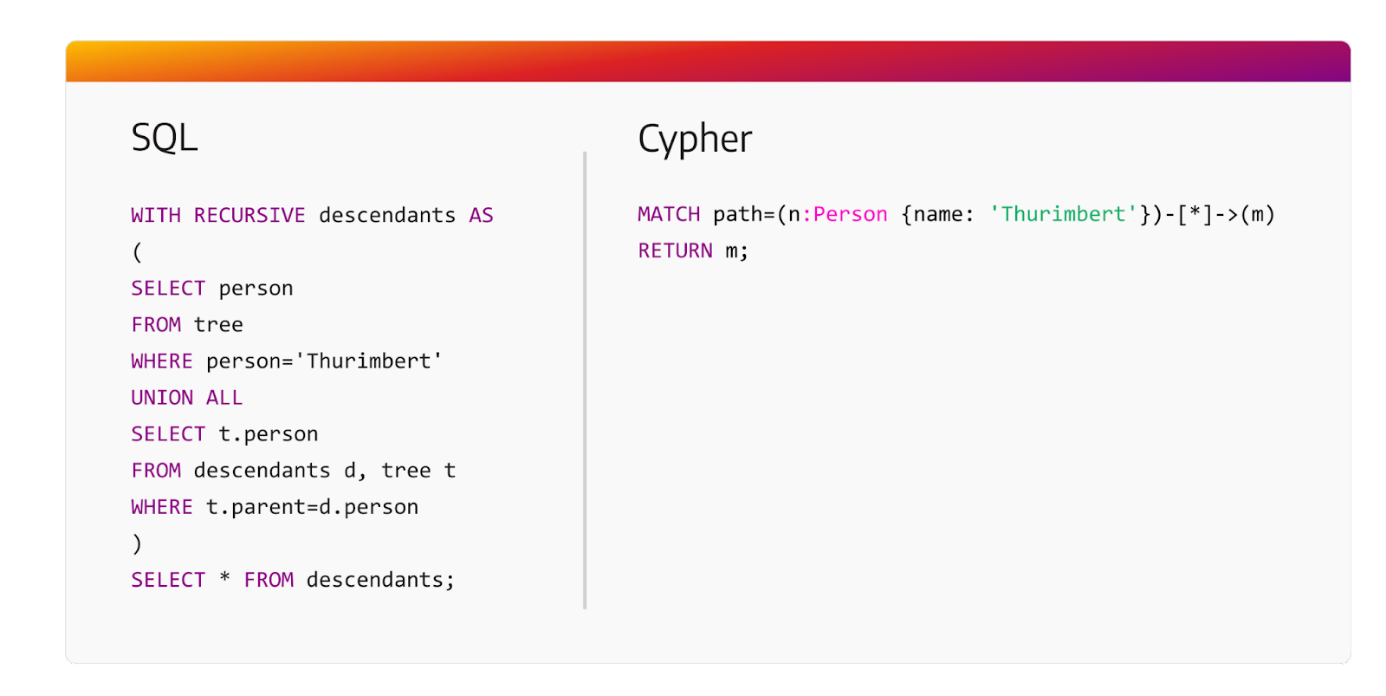
WHERE movies.title = "The Matrix"

**Cypher:**

MATCH (actor:Actor)-[:ACTED\_IN]->(movie:Movie {title: 'The Matrix'})

RETURN actor.name

Konu ile ilgili diğer bir önek ise aşağıdaki görseldeki gibi bir sorgunun hem SQL hem de Cypher karşılıklarını inceleyebilirsiniz.



SQL’e aşina olan herkes Cypher’ı kolaylıkla anlayacaktır. Her ikisi de deklaratif query dilleridir. Cypher, özellikle graphlar için tasarlanmıştır. Her ikisi de clauses, keywordler, expressionlar, operatörler ve functionlar içerir. SQL’den farklı olarak Cypher graph pattern’i merkezine alır. Cypher ile de kompleks sorgular yazabilirsiniz. Cypher, karmaşık okuma ve yazma operasyonlarını tek bir state üzerinden gerçekleştirir.

**Sorguları incelemeye devam etmek için:** [**https://neo4j.com/docs/cypher-manual/current/queries/**](https://neo4j.com/docs/cypher-manual/current/queries/)

**KAYNAKÇA**

[**https://neo4j.com/docs/cypher-manual/current/introduction/cypher-overview/**](https://neo4j.com/docs/cypher-manual/current/introduction/cypher-overview/)

[**https://en.wikipedia.org/wiki/Cypher\_(query\_language)**](https://en.wikipedia.org/wiki/Cypher_(query_language))

[**https://neo4j.com/docs/getting-started/cypher-intro/cypher-sql/**](https://neo4j.com/docs/getting-started/cypher-intro/cypher-sql/)

[**https://medium.com/@fatihserhatturan/neo4j-ile-graf-veritabanlar%C4%B1-uygulamal%C4%B1-bir-rehber-184b95e8c0ca**](https://medium.com/@fatihserhatturan/neo4j-ile-graf-veritabanlar%C4%B1-uygulamal%C4%B1-bir-rehber-184b95e8c0ca)