***Neo4j:***

**İster 1:** Ben Gülnur Yıldız, ASELSAN’da çalışan Ahmet Doğan adlı kişiye nasıl ulaşabilirim?

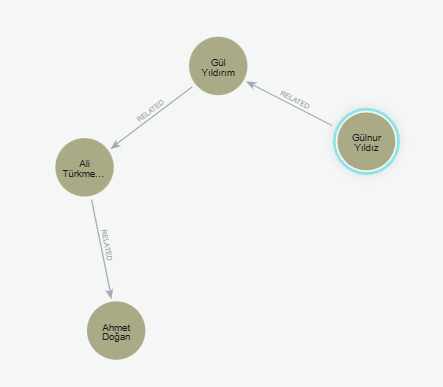
**Sorgu:**

MATCH p=(kaynak:Person {ad: "Gülnur Yıldız"})-[:RELATED\*1..3]->(hedef:Person {ad: "Ahmet Doğan"})

WHERE "ASELSAN" IN hedef.calistigi\_kurumlar

RETURN p LIMIT 1

**Output:**



**İster 2:** Ben Nefise Aydın, ilkokul arkadaşlarımın tüm üniversite arkadaşlarını bulmak istiyorum.

**Sorgu:**

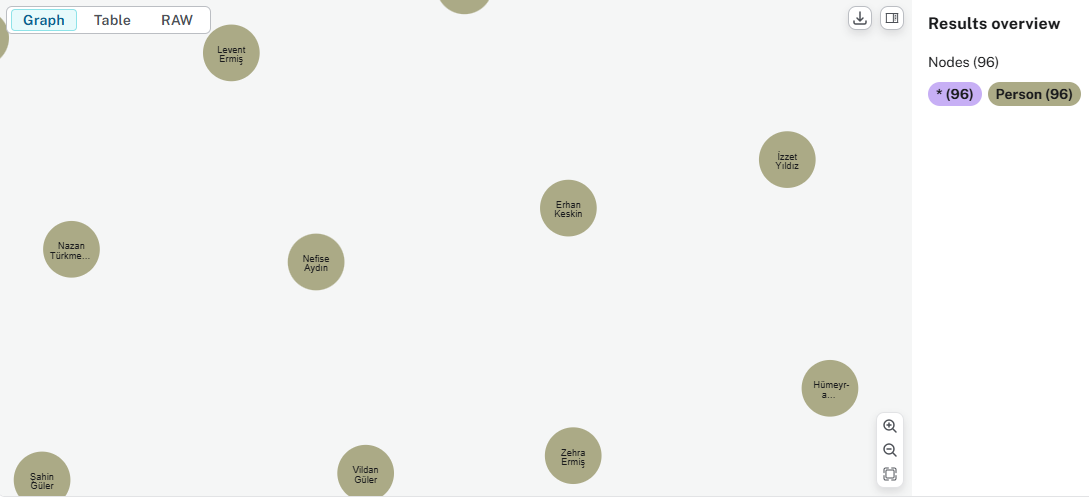
MATCH (kaynak:Person {ad: "Nefise Aydın"})-[:RELATED]->(ara:Person)-[:RELATED]->(hedef:Person)

WHERE ara.ilkokul = kaynak.ilkokul

AND hedef.universite = ara.universite

RETURN hedef

**Output:**

****

(count methodu ile 91 kişi olduğunu görebiliriz.)

**İster 3:** Ben İzzet Yıldız, hemşehrilerimin şu an nerelerde çalıştığını bulmak istiyorum.

**Sorgu:**

MATCH (kaynak:Person {ad: "İzzet Yıldız"})-[:RELATED]-(hedef:Person)

WHERE kaynak.dogum\_yeri = hedef.dogum\_yeri

RETURN DISTINCT hedef.ad, hedef.calistigi\_kurumlar[-1]

**Output:**



(Şu anda çalıştığı yeri bulmak daha mantıklı geldiği için [-1] index i koydum.)

**İster 4:** Ben Emrah Baran, yüksek lisans arkadaşlarımın çocuklarını listelemek istiyorum.

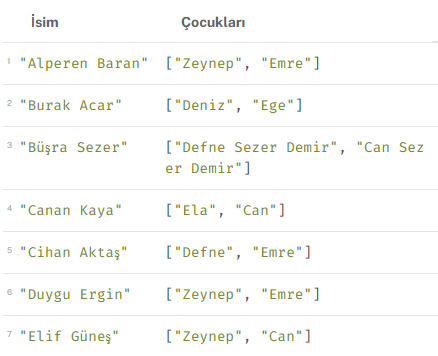
**Sorgu:**

MATCH (kaynak:Person {ad: "Emrah Baran"})-[:RELATED]->(hedef:Person)

WHERE kaynak.yuksek\_lisans = hedef.yuksek\_lisans

RETURN DISTINCT hedef.ad AS İsim, hedef.cocuklar AS Çocukları

**Output:**



**İster 5:** Ben Ali Türkmen, aynı şirkette çalıştığım kişilerin şu an hangi şehirde yaşadığını ve kaç yıl birlikte çalıştığımızı öğrenmek istiyorum.

**Sorgu:**

MATCH (kaynak:Person {ad: "Ali Türkmen"})-[:RELATED]-(hedef:Person)

WITH kaynak, hedef,

apoc.coll.zip(kaynak.calistigi\_kurumlar, kaynak.calisma\_yillari) AS kaynak\_kurum\_yillar,

apoc.coll.zip(hedef.calistigi\_kurumlar, hedef.calisma\_yillari) AS hedef\_kurum\_yillar

UNWIND kaynak\_kurum\_yillar AS kaynak\_entry

UNWIND hedef\_kurum\_yillar AS hedef\_entry

WITH hedef,

kaynak\_entry[0] AS kurum1, kaynak\_entry[1] AS yil1,

hedef\_entry[0] AS kurum2, hedef\_entry[1] AS yil2

WHERE kurum1 = kurum2

WITH DISTINCT hedef, kurum1 AS ortak\_kurum, yil1, yil2,

split(yil1, "-")[0] AS b1, split(yil1, "-")[1] AS e1,

split(yil2, "-")[0] AS b2, split(yil2, "-")[1] AS e2

WITH hedef, ortak\_kurum,

toInteger(apoc.coll.max([b1, b2])) AS ortak\_baslangic,

toInteger(apoc.coll.min([e1, e2])) AS ortak\_bitis

WHERE ortak\_bitis >= ortak\_baslangic

WITH DISTINCT

hedef.ad AS İsim,

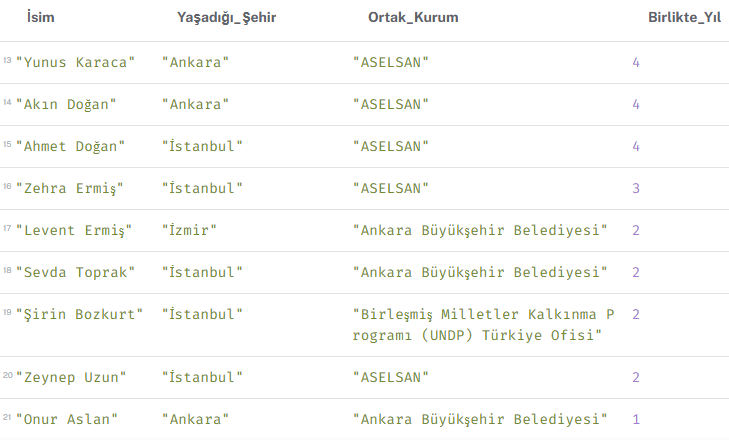
hedef.yasadigi\_sehir AS Yaşadığı\_Şehir,

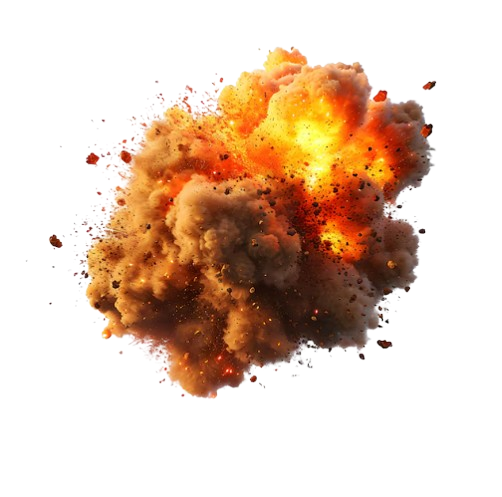
ortak\_kurum AS Ortak\_Kurum,

(ortak\_bitis - ortak\_baslangic + 1) AS Birlikte\_Yıl

RETURN İsim, Yaşadığı\_Şehir, Ortak\_Kurum, Birlikte\_Yıl

ORDER BY Birlikte\_Yıl DESC



TÜM RUNTIME’LARIN KARSILASTIRILMASI

|  |  |  |
| --- | --- | --- |
| İster | Neo4j | mongodb |
| İster 1 | 0.039 saniye | 5.939 saniye |
| İster 2 | 0.156 saniye | 0.053 saniye |
| İster 3 | 0.011 saniye | 0.073 saniye |
| İster 4 | 0.006 saniye | 0.051 saniye |
| İster 5 | 0.072 saniye | 0.083 saniye |