**NEO4J CYPHER SORGULARI İLE CSV YÜKLEME**

Bu haftanın ödevinde kişiler ve icra dosyaları arasındaki ilişkileri csv dosyaları kullanarak modelledim. Csv dosyalarından veri yükleyerek kişilerin özelliklerini, icra dosyalarının detaylarını, hangi dosyalarda alacaklı, hangilerinde borçlu sıfatıyla bulunduklarını cypher sorguları ile oluşturdum.

1. **Veri Yapıları**
   1. **Kişiler –** TC No, İsim, Doğum Yılı, Yaş, Şehir, Meslek, İşyeri Ünvanı
   2. **Alacaklılar –** Alacaklı Id, Alacaklı Tipi (Kişi/Firma), Alacaklı İsmi, Bulunduğu Şehir
   3. **İcra Dosyaları –** Dosya No, Açılış Tarihi, Şehir, Alacaklı Id, Borçlu TC No, Borç Miktarı, Durum (Açık/Kapalı)
   4. **Borçlular –** TC No, İsim, Şehir, Doğum Yılı
2. **Verileri Oluşturma**
   1. **Constraint Tanımlama**

Veri tutarlılığını sağlamak ve kendini tekrarlayan verilerin önüne geçmek için unique key oluşturdum:

CREATE CONSTRAINT tc\_no\_unique IF NOT EXISTS FOR (k:Kisi) REQUIRE k.tc\_no IS UNIQUE;

CREATE CONSTRAINT dosya\_no\_unique IF NOT EXISTS FOR (i:İcra) REQUIRE i.dosya\_no IS UNIQUE;

CREATE CONSTRAINT alacakli\_id\_unique IF NOT EXISTS FOR (a:Alacakli) REQUIRE a.alacakli\_id IS UNIQUE;

* 1. **Csv dosyalarından verileri yükleme ve ilişkileri oluşturma**

LOAD CSV WITH HEADERS FROM 'https://drive.google.com/uc?export=download&id=1Yly996GNRdBA1kt8oFRr9D16fhUZBDBO' AS row

CREATE (k:Kisi {

tc\_no: row.tc\_no,

isim: row.isim,

dogum\_yili: toInteger(row.dogum\_yili),

yas: toInteger(row.yas),

sehir: row.sehir,

meslek: row.meslek,

isyeri\_unvani: row.isyeri\_unvani

});

LOAD CSV WITH HEADERS FROM 'https://drive.google.com/uc?export=download&id=1do4MAALqv1kzZb4YUFeGacybd0OVE5O7' AS row

CREATE (a:Alacakli {

alacakli\_id: toInteger(row.alacakli\_id),

alacakli\_tipi: row.alacakli\_tipi,

isim: row.isim,

sehir: row.sehir

});

LOAD CSV WITH HEADERS FROM 'https://drive.google.com/uc?export=download&id=1-TzkcyjH6iNh6AYBg3-3ZtnLpFmqb9JV' AS row

MERGE (b:Kisi {tc\_no: row.tc\_no})

ON CREATE SET

b.isim = row.isim,

b.sehir = row.sehir,

b.dogum\_yili = toInteger(row.dogum\_yili);

LOAD CSV WITH HEADERS FROM 'https://drive.google.com/uc?export=download&id=1h7h1dtOzDaih\_81XX5o6CtwLBBBsXSiF' AS row

CREATE (i:İcra {

dosya\_no: row.dosya\_no,

acilis\_tarihi: date(row.acilis\_tarihi),

sehir: row.sehir,

borc\_miktari: toFloat(row.borc\_miktari),

durum: row.durum

});

LOAD CSV WITH HEADERS FROM 'https://drive.google.com/uc?export=download&id=1h7h1dtOzDaih\_81XX5o6CtwLBBBsXSiF' AS row

MATCH (i:İcra {dosya\_no: row.dosya\_no})

MATCH (a:Alacakli {alacakli\_id: toInteger(row.alacakli\_id)})

CREATE (a)-[:ALACAKLI\_OLDUGU]->(i);

LOAD CSV WITH HEADERS FROM 'https://drive.google.com/uc?export=download&id=1h7h1dtOzDaih\_81XX5o6CtwLBBBsXSiF' AS row

MATCH (i:İcra {dosya\_no: row.dosya\_no})

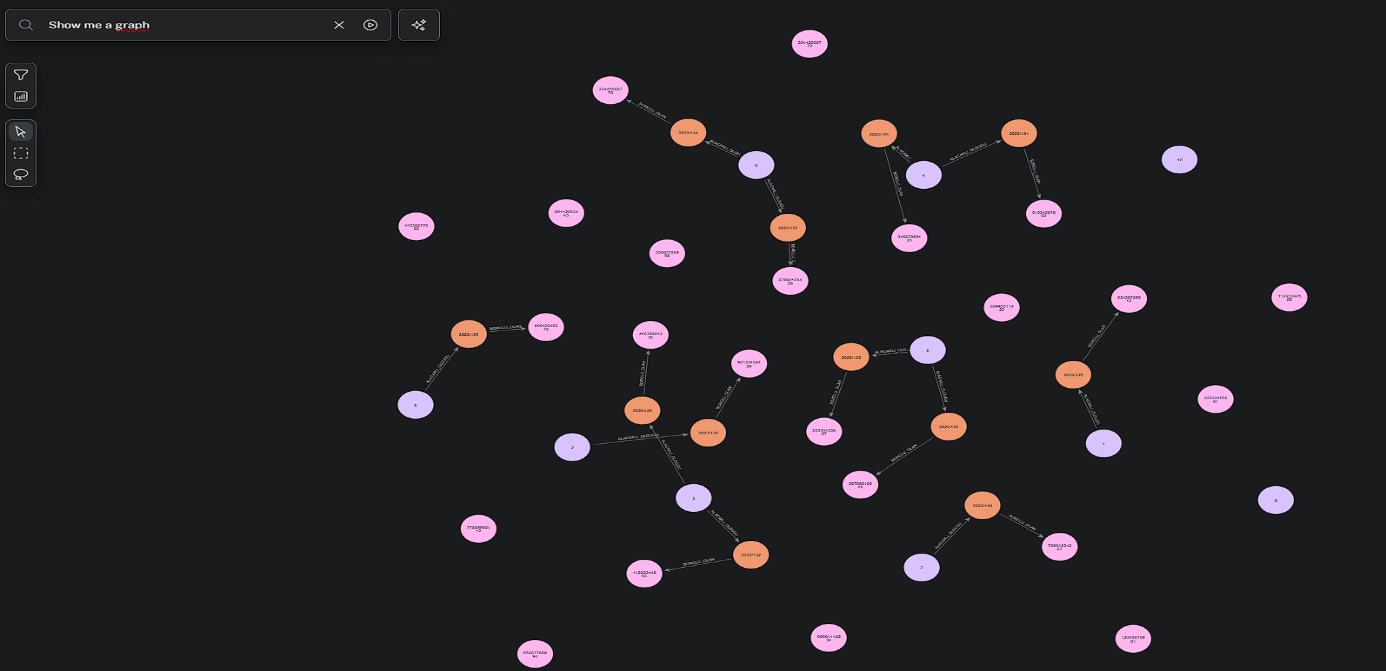
MATCH (b:Kisi {tc\_no: row.borclu\_tc})

CREATE (i)-[:BORCLU\_OLAN]->(b);

Bu şekilde cypher sorguları ile mevcut csv dosyalarından Kisiler, Alacaklılar, İcra Dosyaları oluşturuldu ve birbirleri arasında ALACAKLI\_OLDUGU ve BORCLU\_OLAN ilişkileri de sağlanmış oldu.

1. **Sorgu Örnekleri**

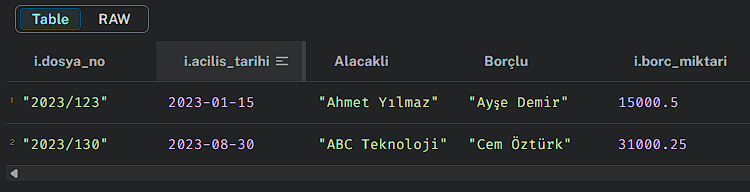
Tüm veriler yüklendiğindeki graph görüntüsü şu şekildedir:



Belirli bir şehirdeki icra dosyalarını görüntülemek istersek:

**MATCH (i:İcra {sehir: "İstanbul"})<-[:ALACAKLI\_OLDUGU]-(a)**

**RETURN i.dosya\_no, a.isim, i.borc\_miktari;**

****

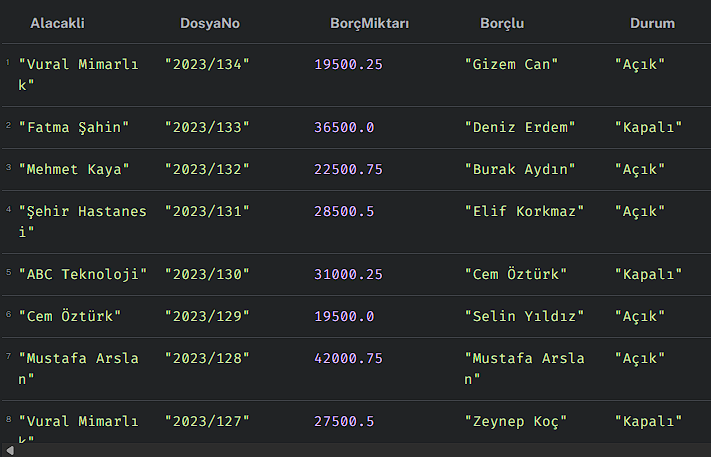
Tüm icra dosyaları ve detaylarını görüntüleme:

**MATCH (a:Alacakli)-[:ALACAKLI\_OLDUGU]->(i:İcra)-[:BORCLU\_OLAN]->(b:Kisi)**

**RETURN a.isim AS Alacakli, i.dosya\_no AS DosyaNo, i.borc\_miktari AS BorçMiktarı,**

**b.isim AS Borçlu, i.durum AS Durum**

**ORDER BY i.acilis\_tarihi DESC;**

****