NewMind AI Bootcamp 4.Hafta Ödevi

Adı-Soyadı: Yunus BİLGİÇ

## Veri Setinin Özellikleri

Bu çalışmada Kaggle platformunda yayımlanan "Online Retail Transaction Data" veri seti kullanılmıştır. Veriler, müşteri işlemleri, ürün bilgileri, fatura numaraları ve ülkeler gibi çeşitli alanları içermektedir. Veri setlerinin isimleri şu şekildedir:

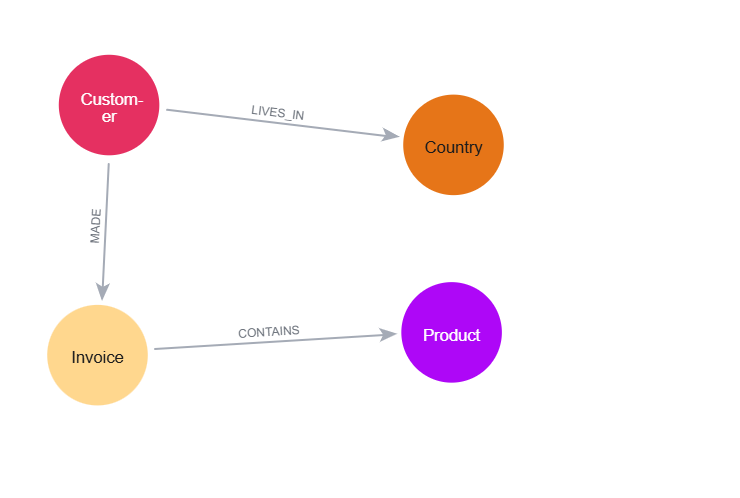
* Invoices.csv
* Products.csv
* Invoice\_rel\_product.csv
* Customers.csv

Veri setlerinde kullanılan sütunların anlamı şu şekildedir:

* ***InvoiceNo:*** Her siparişe ait fatura numarası.
* ***InvoiceDate:*** Fatura kesilme tarihi.
* ***StockCode:*** Her ürüne ait stok kodu.
* ***Description:*** Her ürüne ait tanım bilgileri.
* ***UnitPrice:*** Her ürünün birim fiyat bilgisi.
* ***Quantity:*** Belirli bir üründen kaç adet alındığının bilgisi.
* ***CustomerId:*** Müşteri id’si.
* ***Country:*** Müşterinin hangi ülkeden sipariş verdiğinin bilgisi.
* Veri setinde InvoiceNo sütununda “C” ile başlayan faturalar iade edilen faturaları belirtmek için kullanılmış. Bu farkı belirtmek için ben de graph veri tabanını aktarırken bu durumu bir özellik olarak ekledim. {is\_return:True/False}
* Bazı satırlarda UnitPrice ve Quantity negatif değerler alıyordu. Bu değerlerin iadeler olup olmadığını sorguladığımda iadeler olduğunu gördüm. Bundan dolayı Pandas Dataframe’i ile o satırları (-1) ile çarparak pozitif hale getirdim. Çünkü öncesinde {is\_return:True/False} propertysi eklemiştim.
* CustomerId ve description sütunlarındaki eksik değerleri yine Pandas ile temizledim.
* Country bölümü müşterilerin bir property si olarak da verilebilirdi, ***fakat ülkeler çok sık kullanıldığı için*** yeni bir node oluşturmanın daha doğru olacağını düşündüm.

## Node ve İlişki Yapılarının Görsel Gösterimi

* CALL db.schema.visualization



## Node ve İlişki Yapılarının Neo4j Ortamına Aktarılması

**// Müşteriler için CustomerID, ürünler için StockCode,**

**// faturalar için InvoiceNo, ülkeler için name özellikleri eşsiz hale getirildi.**

CREATE CONSTRAINT customer\_id\_unique IF NOT EXISTS

FOR (c:Customer) REQUIRE c.customer\_id IS UNIQUE

CREATE CONSTRAINT product\_code\_unique IF NOT EXISTS

FOR (p:Product) REQUIRE p.stock\_code IS UNIQUE

CREATE CONSTRAINT invoice\_no\_unique IF NOT EXISTS

FOR (i:Invoice) REQUIRE i.invoice\_no IS UNIQUE

CREATE CONSTRAINT country\_name\_unique IF NOT EXISTS

FOR (co:Country) REQUIRE co.name IS UNIQUE

**// Doğrudan Kaggle’dan veri çekilemediği için temizlediğim dataframeleri kendi Github repoma //yükledim. Ve verileri oradan yükledim.**

LOAD CSV WITH HEADERS FROM 'https://raw.githubusercontent.com/simulakr/Online-Retail-Transactions/main/customers.csv' AS customer

MERGE (c:Customer {customer\_id: customer.CustomerID})

MERGE (co:Country {name: customer.Country})

MERGE (c)-[:LIVES\_IN]->(co)

LOAD CSV WITH HEADERS FROM ‘https://raw.githubusercontent.com/simulakr/Online-Retail-Transactions/main/products.csv’ AS product

MERGE (p:Product {stock\_code: product. StockCode })

SET p.description = product.Description,

p. unitprice = toFloat(product. UnitPrice)

**// InvoiceDate yüklenirken bazı hatalar ile karşılaştım. Bunun çözümü için ‘apoc.date.parse’ metodunu kullandım. Customer ile invoice ilişkisini kurarken nodelarımı ‘With’ ile çektim.**

LOAD CSV WITH HEADERS FROM ‘https://raw.githubusercontent.com/simulakr/Online-Retail-Transactions/main/invoices.csv’ AS invoice

MERGE (i:Invoice {invoice\_no: invoice. InvoiceNo })

MERGE (i:Invoice {invoice\_no: invoice.InvoiceNo})

SET i.invoiceDate = datetime({epochMillis: apoc.date.parse(invoice.InvoiceDate, 'ms', 'M/d/yyyy H:mm')}),

i.is\_return = invoice.is\_return

WITH invoice, i

MATCH (c:Customer {customer\_id: invoice.CustomerID})

MERGE (c)-[:MADE]->(i);

LOAD CSV WITH HEADERS FROM ‘https://raw.githubusercontent.com/simulakr/Online-Retail-Transactions/main/invoice\_rel\_product.csv’ AS rel

MATCH (i:Invoice {invoice\_no: rel. InvoiceNo })

MATCH (p:Product {stock\_code: rel. StockCode })

MERGE (i)-[r:CONTAINS]->(p)

SET r.quantity = toInteger(rel.Quantity)