­

Big Data Aplicado

Práctica Neo4J

En este documento tendréis que elaborar un documento, a modo de tutorial, que incluya toda la información (código, capturas de pantalla, resultados). En el aula virtual subid el documento en formato PDF.

Índice

[Importación de datos 3](#_Toc183689400)

[Aplicación de viajes 6](#_Toc183689401)

[Consultaría empresa de viajes 6](#_Toc183689402)

# Importación de datos

1. **Escribir el comando de importación, recomendable usar LOAD\_CSV.**

Tras descargar el archivo, creamos una carpeta y dentro ponemos el .tar

Posteriormente abrimos cmd y ponemos el comando **tar -xvf yelp\_dataset.tar** para descomprimir

Después, lo pasamos a CSV utilizando [esta herramienta](https://github.com/Yelp/dataset-examples/blob/master/json_to_csv_converter.py) (hay que cambiar el Python 2 a 3)

Una vez pasado a CSV, abrimos la PowerShell y lo cortamos con

**Get-Content "[ruta]\yelp\_academic\_dataset\_business.csv" | select -First 10000 | Out-File "[ruta]\business\_10000.csv"**

Esto devuelve las 10000 primeras líneas

Después, lo pasamos al import de neo4j usando

**docker cp "[ruta]\business\_2000.csv" Neo4j\_DBA:/var/lib/neo4j/import/**

Una vez pasados todos los CSV, hacemos un **docker exec -it Neo4j\_DBA bash**

Y luego **cd /var/lib/neo4j/import** y **ls** para verificar que están ahí.

Después, importamos usando los comandos:

LOAD CSV WITH HEADERS FROM 'file:///business\_2000.csv' AS row

CREATE (n:Business)

set n = row

LOAD CSV WITH HEADERS FROM 'file:///checkin\_2000.csv' AS row

CREATE (n:Checkin)

set n = row

LOAD CSV WITH HEADERS FROM 'file:///tip\_2000.csv' AS row

CREATE (n:Tip)

set n = row

LOAD CSV WITH HEADERS FROM 'file:///user\_2000.csv' AS row

CREATE (n:User)

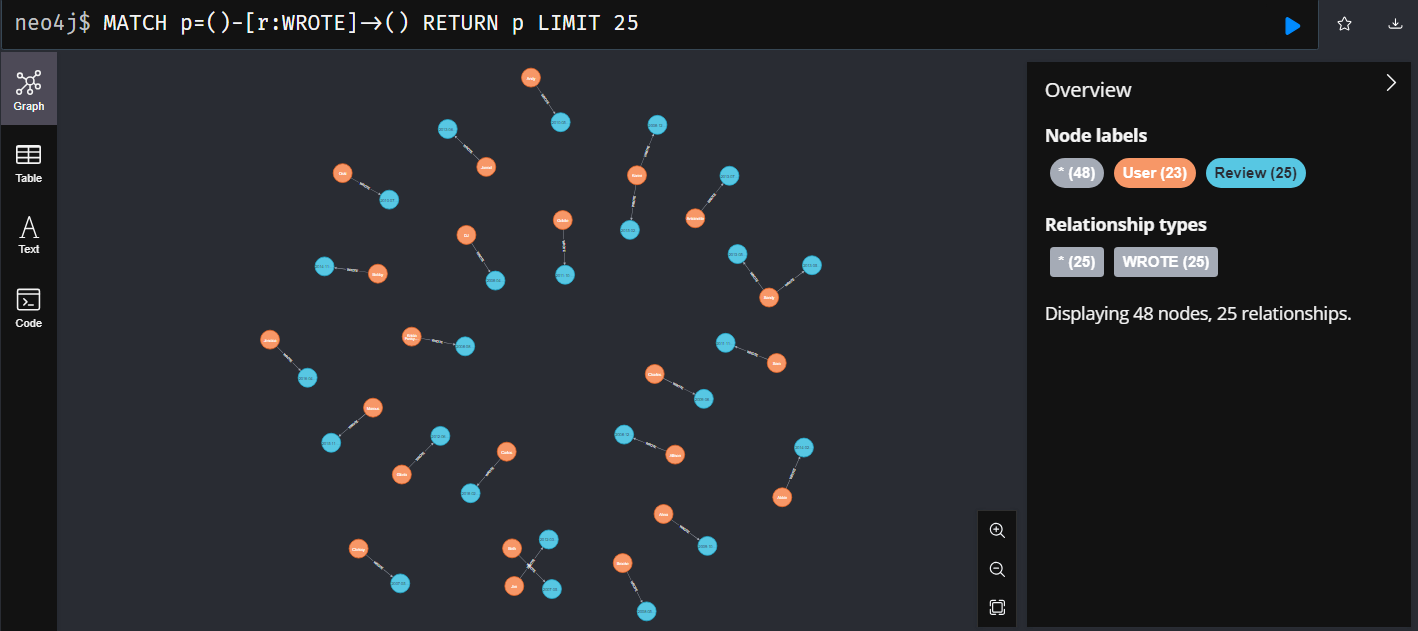
set n = row

LOAD CSV WITH HEADERS FROM 'file:///review\_2000.csv' AS row

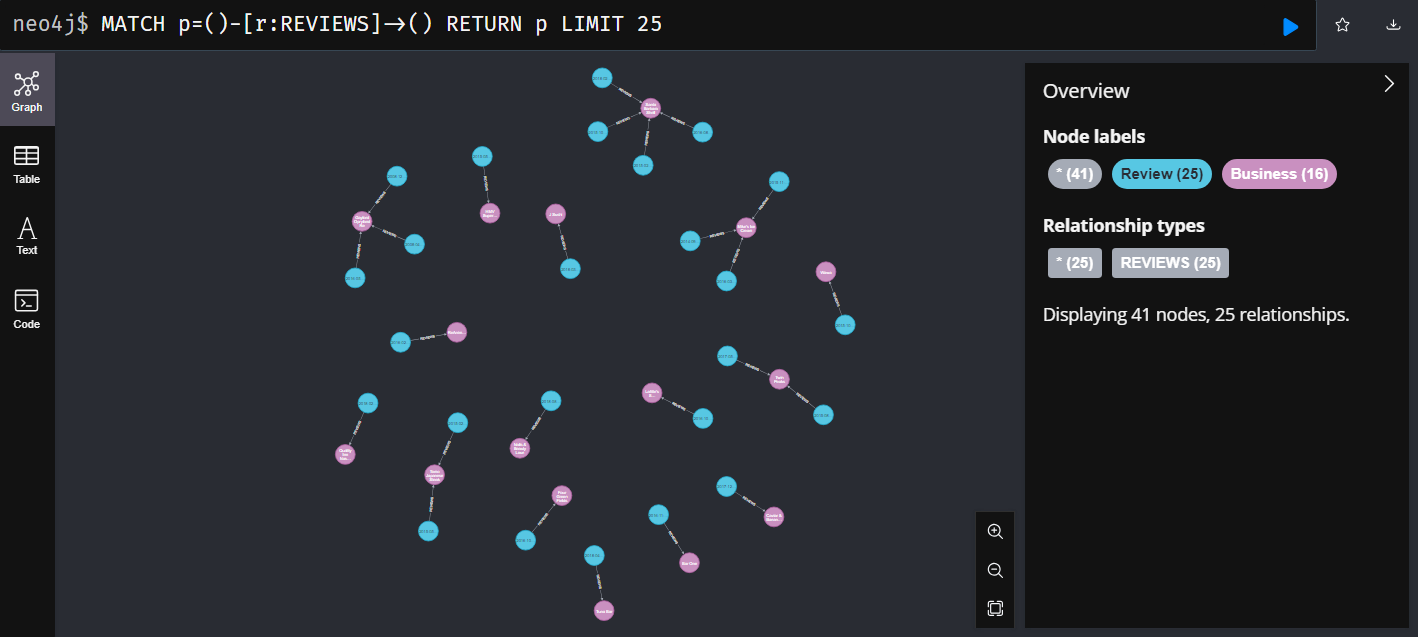
CREATE (n:Review)

set n = row

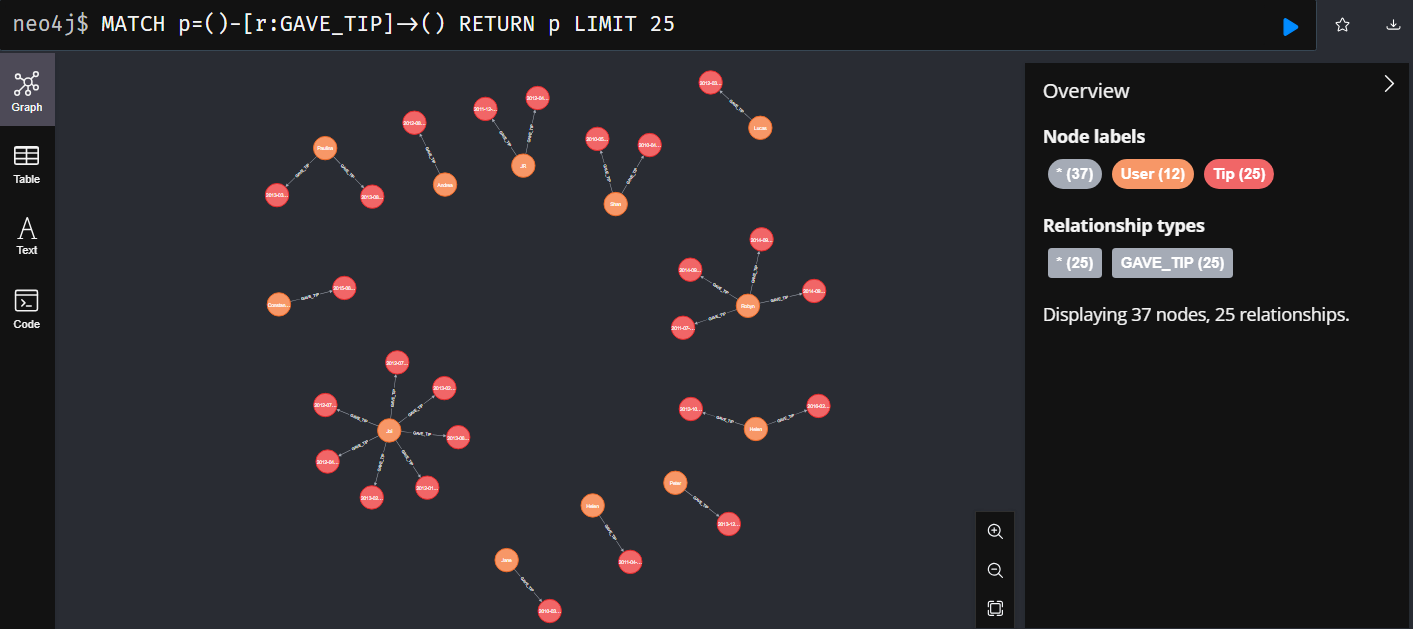
1. **Realizar un grafo del modelo y mostrar captura.**



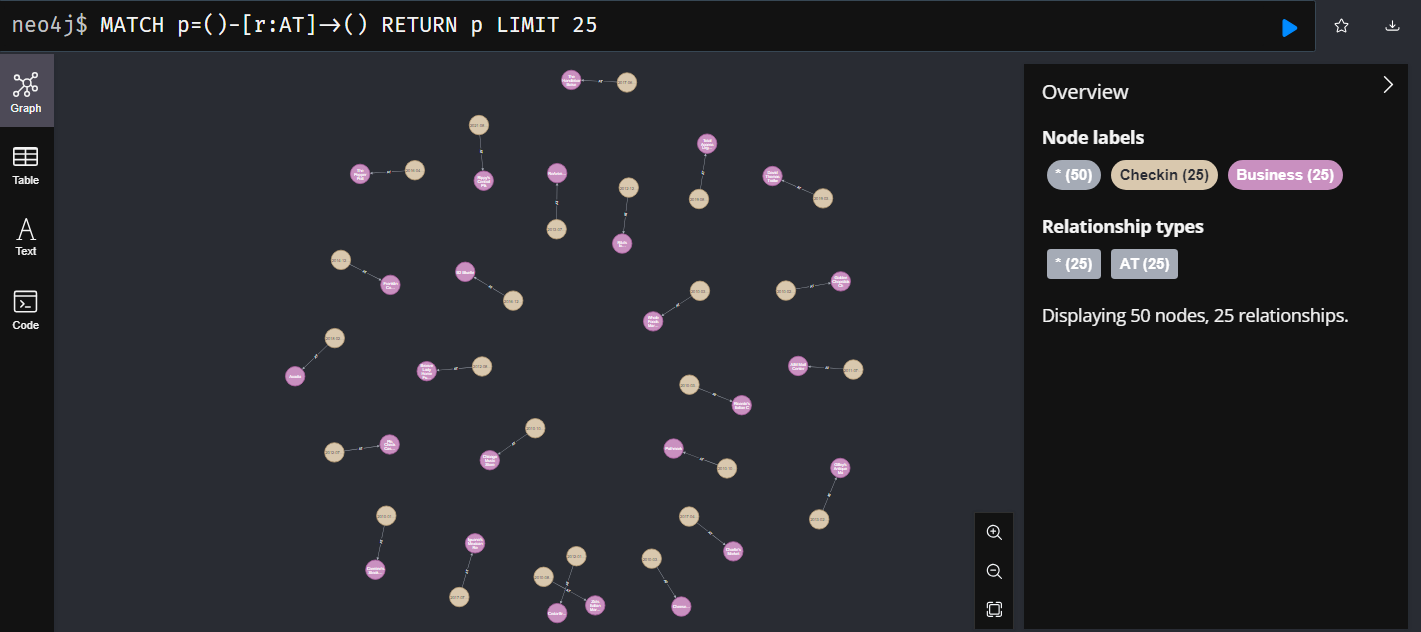
Relación WROTE (User con review)



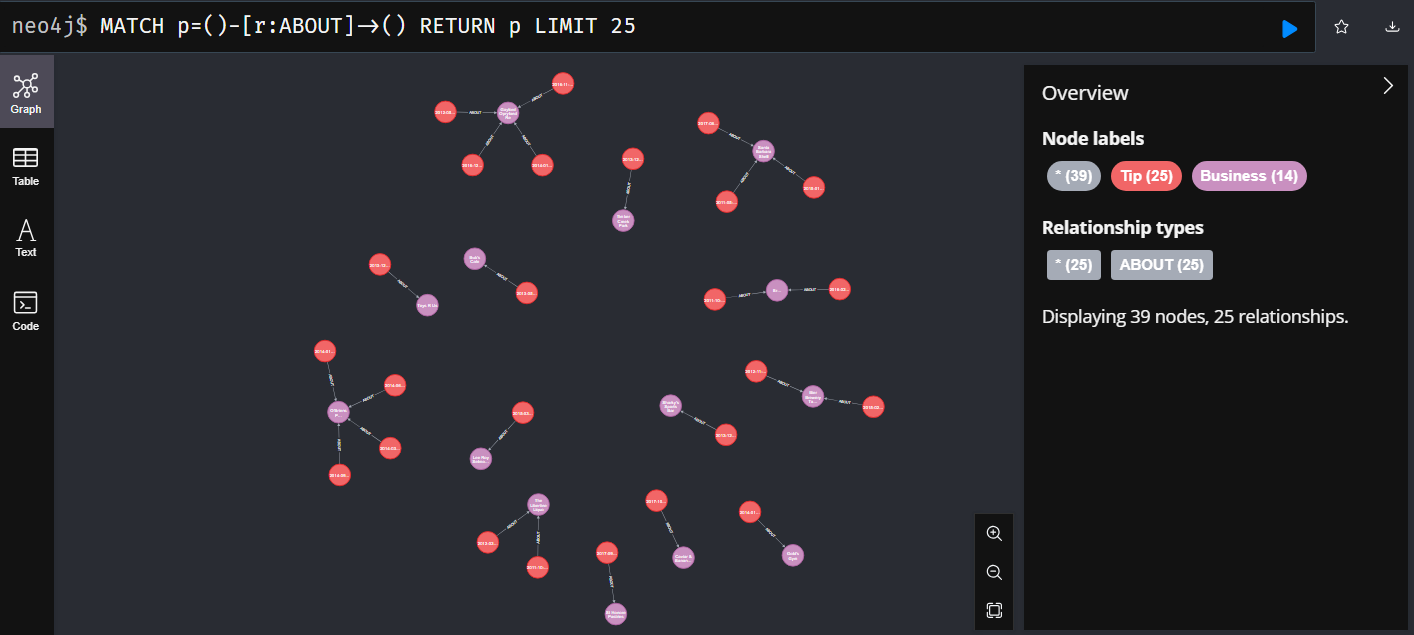
Relación REVIEWS (Review con Business)



Relación GAVE\_TIP (User con Tip)



Relación AT (Checking con Business)



Relación ABOUT (Tip con Business)

1. **Describir los nodos y sus relaciones**

* **Business**: Negocio que ofrece servicios
* **User**: Usuario que necesita un servicio de Business
* **Review**: Valoración que da el user al business
* **Tip**: Propina que da el user por el servicio que ofrece business
* **Checkin**: Si está registrado o no el user en el business

# Aplicación de viajes

1. **Encuentra los 10 hoteles con mayor número de reviews.**

MATCH (b:Business)-[:REVIEWS]-(r:Review)

WHERE b.categories CONTAINS "Hotel"

RETURN b.name AS HotelName, COUNT(r) AS ReviewCount

ORDER BY ReviewCount DESC

LIMIT 10;

1. **Encuentra los 10 usuarios con un número mayor de reviews realizadas**.

MATCH (u:User)-[:WROTE]->(r:Review)

RETURN u.name AS UserName, u.user\_id AS UserID, COUNT(r) AS ReviewCount

ORDER BY ReviewCount DESC

LIMIT 10;

# Consultaría empresa de viajes

1. **Encuentra los 50 usuarios con mayor número de reviews que han hecho  
   una review del hotel Bellagio Hotel**.

MATCH (u:User)-[:WROTE]->(r:Review)-[:REVIEWS]->(h:Business)

WHERE h.name = 'Bellagio Hotel'

RETURN u.name AS usuario, COUNT(r) AS totalReviews

ORDER BY totalReviews DESC

LIMIT 50;

1. **Buscar todos los hoteles que estos usuarios han hecho una review, no vale  
   introducir los usuarios en el Where de manera manual**.

MATCH (u:User)-[:WROTE]->(r:Review)-[:REVIEWS]->(h:Business)

WHERE h.name = 'Bellagio Hotel'

WITH u

MATCH (u)-[:WROTE]->(r2:Review)-[:REVIEWS]->(h2:Business)

RETURN DISTINCT h2.name AS hotel

ORDER BY hotel;

1. **Obtén el hotel con mayor número de reviews de los usuarios obtenidos en  
   el apartado 1 de esta sección**

MATCH (u:User)-[:WROTE]->(r:Review)-[:REVIEWS]->(h:Business)

WHERE h.name = 'Bellagio Hotel'

WITH u

MATCH (u)-[:WROTE]->(r2:Review)-[:REVIEWS]->(h2:Business)

RETURN h2.name AS hotel, COUNT(r2) AS totalReviews

ORDER BY totalReviews DESC

LIMIT 1;