

**Máster universitario de Ciencia de Datos**

**Práctica 1 – PRA1**

**Uso de base de datos NoSQL - Cassandra, MongoDB y Neo4j.**

Autor:

Mario Ubierna San Mamés

|  |
| --- |
| Índice de Contenido |

[Índice de Contenido 3](#_Toc89849666)

[Índice de tablas 4](#_Toc89849667)

[Índice de ilustraciones 5](#_Toc89849668)

[1. Ejercicio 1: Cassandra 6](#_Toc89849669)

[2. Ejercicio 2: MongoDB 7](#_Toc89849670)

[3. Ejercicio 3: Neo4j 8](#_Toc89849671)

[4. Ejercicio 4: Neo4j 9](#_Toc89849672)

[5. Bibliografía 10](#_Toc89849673)

|  |
| --- |
| Índice de tablas |

|  |
| --- |
| Índice de ilustraciones |

|  |
| --- |
| Ejercicio 1: Cassandra |

A

|  |
| --- |
| Ejercicio 2: MongoDB |

A

|  |
| --- |
| Ejercicio 3: Neo4j |

*Para la realización de este ejercicio se seguirán las instrucciones del caso de estudio ubicado en la siguiente URL: https://neo4j.com/developer/guide-importing-data-and-etl/ . Este caso se aborda también en el documento de ejercicios titulado “Transformación de una base de datos relacional a un modelo en grafo” pero a un nivel más superficial. Se recomienda leer dicho documento antes de realizar este ejercicio.*

*Se recomienda leer con atención la página web, ya que se introducen conceptos y buenas prácticas a seguir en ‘production’.*

*En la máquina virtual ya hay una base de datos precargada (twitter) que no es necesaria para realizar los ejercicios de esta práctica. Los nodos y relaciones que hay creados no interfieren en el ejercicio, los podéis ignorar tranquilamente.*

*Encontraréis las instrucciones para arrancar el servicio de Neo4j en la página 11 del documento con nombre “Uso de máquina virtual\_ Bases de datos no convencionales.pdf”. Recordad que el usuario / contraseña para acceder a Neo4j son: neo4j / uoc.*

## Ejercicio 3.1 (no puntúa)

Mirar el enunciado de la práctica (carga de los datos).

## Ejercicio 3.2 (30%)

*Se propone responder las siguientes cuestiones:*

*1. Una vez realizada la carga de datos, utilizando cypher y su interfaz web, se pide:*

*a. obtener una visualización del esquema de los nodos/relaciones que se han creado.*

*b. contar cuántos nodos de tipo Supplier se han creado (adjuntar una captura de pantalla de las query y resultado).*

*2. Explicar brevemente (máx 3 líneas) la funcionalidad implementada de la línea:*

*ON CREATE SET c.categoryName = row.CategoryName, c.description = row.Description;*

*3. En la página web, se crea un índice para cada tipo de nodo y una restricción de unicidad para los nodos de tipo Order. Explicar brevemente (máx 3 líneas): ¿Por qué se recomienda crear un índice para cada tipo de nodo?*

### Comprobaciones de la carga de datos

#### Obtener una visualización del esquema

Para obtener una visualización del esquema de los nodos/relaciones que se han creado usamos “*call db.schema,visualization()*”.



Ilustración 1 – función para obtener el esquema.

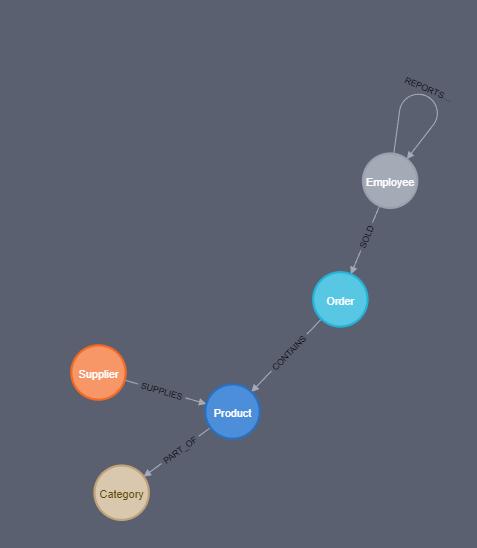


Ilustración 2 - Visualización del esquema.

#### Cuántos nodos tipo *Supplier* se han creado

Para obtener el número de nodos hay que ejecutar la siguiente consulta y obtendremos dicho resultado:



Ilustración 3 - Consulta para obtener el número de nodos de tipo Supplier.

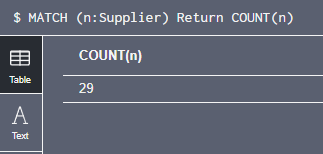
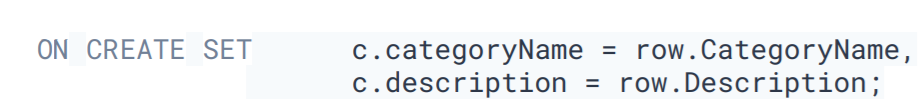


Ilustración 4 - Resultado del número de nodos de tipo Supplier.

### Explicar la funcionalidad implementada



Lo que se está haciendo es definir los valores de las dos propiedades que tiene cada nodo del tipo *Category*, es decir, por cada línea del fichero CSV leído, coge los valores de las columnas (*CategoryName* y *Description*) y los añade a las propiedades correspondientes.

### ¿Por qué se recomienda crear un índice para cada tipo de nodo?

Básicamente se recomienda crear un índice para cada tipo de nodo para garantizar la búsqueda de nodos de forma óptima, dando así un valor único para cada nodo del mismo tipo.

## Ejercicio 3.3 (30%)

*Se pide borrar todos los nodos de tipo Supplier: Adjuntar una captura de pantalla con las queries utilizadas para eliminar todos estos nodos. Ayuda: no es posible borrar nodos que tengan relaciones con otros.*

Para resolver este ejercicio se puede hacer de dos formas, la primera es eliminando primero las relaciones y luego los nodos de tipo *Supplier* (2 consultas), y la segunda es eliminar directamente los nodos pero haciendo uso de la palabra reservada *DETACH* (elimina primero las relaciones y luego los nodos pero todo de una).

En este caso, como no se indica que no se pueda hacer uso de *DETACH*, se ha elegido esta opción ya que es una consulta menos que hay que hacer, dando lugar al siguiente resultado:



Ilustración - Borrado de los nodos tipo Supplier.

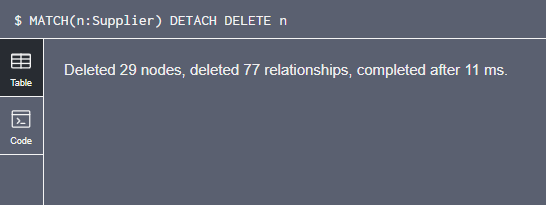


Ilustración - Resultado del borrado de los nodos tipo Supplier.

## Ejercicio 3.4

*Se pide recrear los nodos Supplier y las relaciones SUPPLIES para restablecer el estado de la BBDD y poder responder a las siguientes preguntas:*

*- ¿Cuántos Supplier provienen de Francia? (ayuda: buscar ‘France’ ya que la información es en inglés).*

*- ¿Cuáles son los Supplier cuya web de referencia apunta a una URL absoluta por cada país? Es decir, que contenga una página que empiece por “http”. Se pide identificar cuántos Supplier hay por cada país con una URL absoluta.*

*Se deberá proporcionar las sentencias de creación utilizadas, las dos consultas planteadas y los resultados de las mismas.*

### Recreación de los nodos Supplier y las relaciones SUPPLIES

Lo primero de todo es crear los nodos de tipo *Supplier*, para ello hay que almacenar nuevos campos (*Country, HomePage*) para así satisfacer luego las consultas:

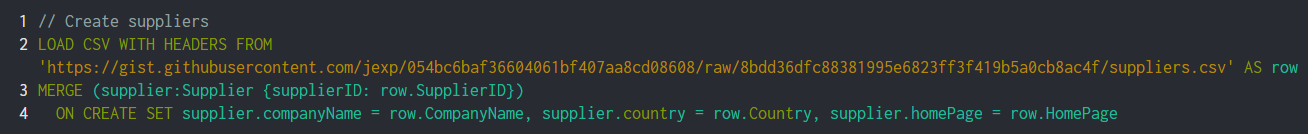


Ilustración - Creación de los nodos de tipo Supplier.

Una vez creado los nodos, creamos las diferentes relaciones, solo se producen con producto:

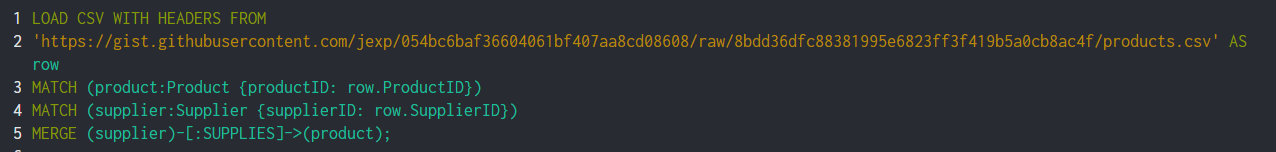


Ilustración - Creación de las relaciones entre Supplier y Product.

### ¿Cuántos Supplier provienen de Francia?

La consulta que tenemos que ejecutar es:



Ilustración - Consulta cuántos Supplier provienen de Francia.

Proporciona el siguiente resultado:

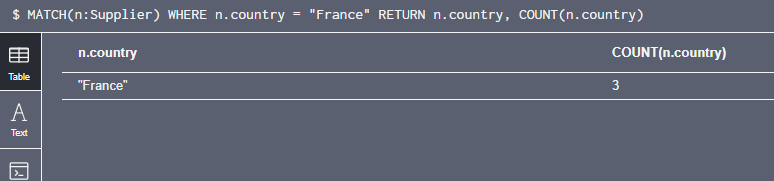


Ilustración - Resultado cuántos Supplier provienen de Francia.

### ¿Cuántos Supplier hay por cada país con una URL absoluta?

La consulta a ejecutar es:



Ilustración - Consulta cuántos países con URL absoluta.

Esta consulta proporciona el siguiente resultado:

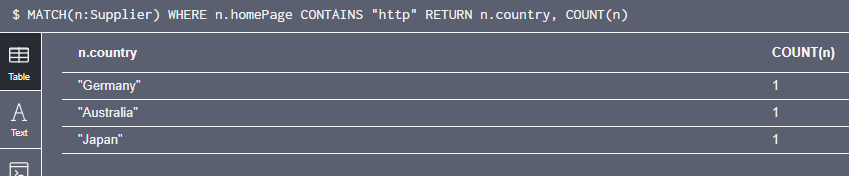


Ilustración - Resultado cuántos países con URL absoluta.

|  |
| --- |
| Ejercicio 4: Neo4j |

*Este ejercicio asume que se ha resuelto el ejercicio anterior. En caso que hayas resuelto el ejercicio 3.B (borrar nodos Supplier) pero no hayas recuperado el estado inicial del DDBB (resuelto el ejercicio 3.C), antes de seguir con este ejercicio crea de nuevo los nodos Supplier y las relaciones SUPPLIES.*

*Se pide proporcionar las siguientes consultas en Cypher y una captura de pantalla con los resultados que se obtienen para las siguientes operaciones:*

## Consulta 1 (20%)

*Encontrar el empleado con título “Sales Representative” que ha vendido (SOLD) más pedidos que contienen la palabra “White”. Listar nombre y apellido del empleado y número de pedidos vendidos.*

La consulta es la siguiente:

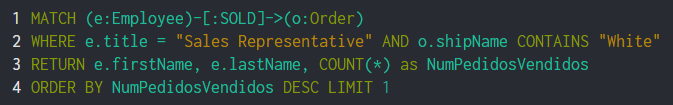


Ilustración – Neo4j consulta 1.

Resultado de la consulta:

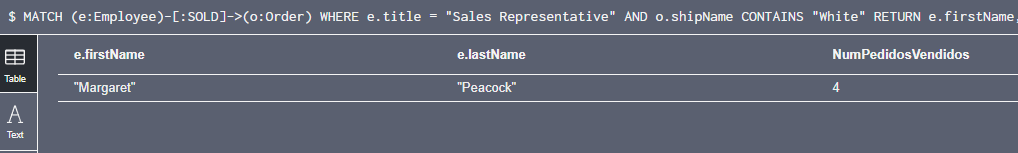


Ilustración - Neo4j resultado consulta 1.

## Consulta 2 (20%)

*Listar el nombre de la categoría asociada (PART\_OF) a 6 productos.*

La consulta realizada es:

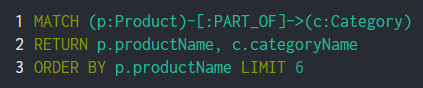


Ilustración - Neo4j consulta 2.

El resultado es:

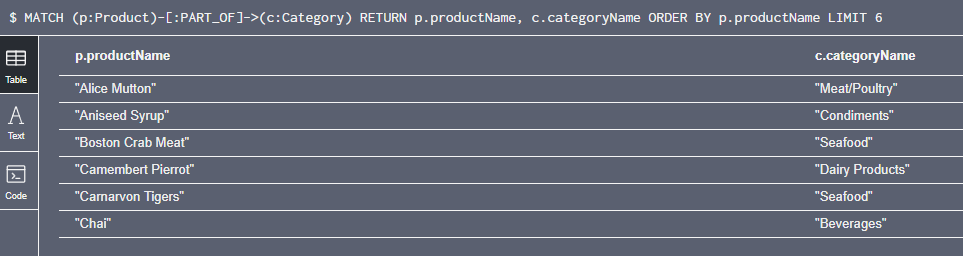


Ilustración - Neo4j resultado consulta 2.

## Consulta 3 (30%)

*Obtener el segundo, tercer y cuarto mejores vendedores de “Tokyo Traders”. Los mejores vendedores son aquellos que han vendido más productos de “Tokyo Traders”. Listar sólo el nombre y apellido de los vendedores y el número de unidades vendidas por vendedor.*

La consulta es:

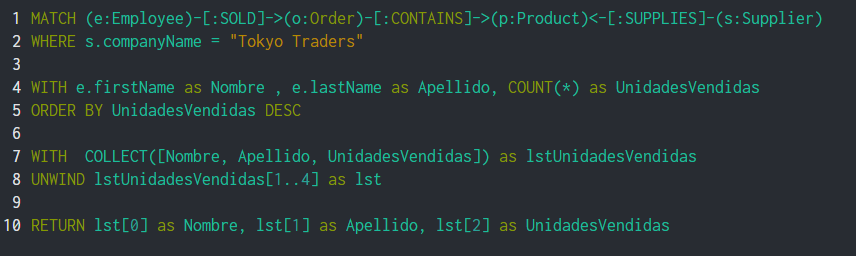


Ilustración - Neo4j consulta 3.

El resultado obtenido:

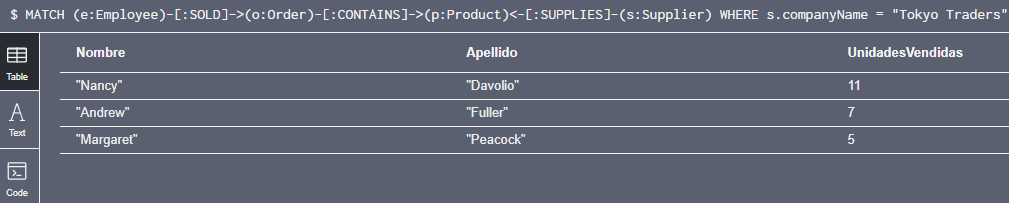


Ilustración - Neo4j resultado consulta 3.

|  |
| --- |
| Bibliografía |

A