**Trabajo Práctico Neo4j – Entorno basado en Docker local**

**Ejercicio 1:** Datos no disponibles en el repo

Ejercicio 1.1. clientes que han dado buenas opiniones sobre el producto cuyo precio son 4000 euros

MATCH (c: Cliente) -[:GOOD]-> (p:Producto {precio: 4000})

RETURN c.nombre;

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ejercicio 1.2. Cadena de apoyos

MATCH path = (c: Cliente)-[:APOYA\*]-(c) RETURN c

Interfaz de usuario gráfica

Descripción generada automáticamente

Ejercicio 1.3. Demetrio apoya a Genara

MATCH (c1: Cliente {nombre: "Demetrio"}), (c2:Cliente {nombre: "Genara"})

MERGE (c1) -[:APOYA]-> (c2)

RETURN c1, c2

Gráfico

Descripción generada automáticamente

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Ejercicio 2. Propuesta de base de datos

**Recetas de cocina y sus maridajes**

He creado una base de datos cuyos nodos son:

* Recetas de cocina, con propiedades { display\_name, tipo\_de\_cocina, minutos\_de\_preparacion y categoría}
* Ingredientes, con propiedades {dislpay\_name y tipo}
* Bebidas, con propiedades {display\_name, tipo y región}

Comentaré el nombre de las propiedades display name en conclusiones.

Las relaciones entre ellos son:

* Marida\_con, con origen en las recetas y destino en las bebidas
* Contiene, con origen en las recetas y destino en los ingredientes

Se ha creado un total de 10 nodos de recetas, 26 de ingredientes y 7 de bebida.

A su vez, hay 30 relaciones de tipo “Contiene” y 11 de tipo “Marida\_con”.

La siguiente imagen ilustra el grafo que representa la base de datos.

Patrón de fondo

Descripción generada automáticamente

**Consultas propuestas:**

1. **Recetas que maridan con bebidas sin alcohol:**

MATCH (r:Receta)-[]->(b:Bebida)

WHERE NOT b.tipo IN ["Vino", "Cerveza"]

RETURN COLLECT(DISTINCT r.display\_name) AS Recetas\_con\_maridaje\_sin\_alcohol;

Se buscan las recetas que maridan con alguna bebida cuyo tipo no sea Vino o Cerveza (bebidas con alcohol). Luego estas recetas se devuelven como una lista por acción de la función COLLECT con el header deseado.

Captura de pantalla de un celular

Descripción generada automáticamente

1. **Lista de maridajes por receta. Se ofrece agua en caso de que no exista ninguna opción**

MATCH (r:Receta)

OPTIONAL MATCH (r)-[:Marida\_con]->(b:Bebida)

RETURN r.display\_name AS Receta, COALESCE(collect(b.display\_name), ["Agua"]) AS Maridaje;

OPTIONAL MATCH permite que las recetas que no cuentan con una relación del tipo “Marida\_con” pasen al RETURN con un valor null (un MATCH no pasaría ningún valor). De esta forma la función COALESCE, al encontrar ese Null le asigna el valor Agua. Los valores devueltos como Maridaje son enlistados por la función COLLECT.

Imagen que contiene Texto

Descripción generada automáticamente

1. **Receta cuyos ingredientes coincide el mayor número de veces con otros platos**

MATCH (r1:Receta)-[:CONTIENE]->(i:Ingrediente)<-[:CONTIENE]-(r2:Receta)

WHERE r1 <> r2

RETURN r1.display\_name AS Receta, COUNT(i) AS Coincidencias

ORDER BY Coincidencias DESC

LIMIT 1;

Se busca las parejas de recetas que comparten un ingrediente y que sean distintas entre sí. Luego se pasan las variables r1 y Coincidencias (el número de ingredientes compartidos por esos dos platos). Finalmente se devuelve el nombre de la Receta y el número de coincidencias con los headers deseados.

Captura de pantalla de un celular

Descripción generada automáticamente

Comentarios

* El nombre display\_name utilizado en las propiedades de los nodos se debe a que Neo4j busca automáticamente que campo mostrar como “nombre” sobre los nodos en los diagramas. Al utilizar la palabra nombre, estaba empleando un campo no deseado para esto y de esta forma forcé el cambio a la propiedad nombre.
* Mis disculpas a los paladares que se puedan ofender con mi ejercicio.