

Tercer trabajo de base de datos 2 (12.5%)

El objetivo de este trabajo es practicar los lenguajes de Neo4j y de MongoDB y aspectos básicos de visualización con Google Colab.

Suponga que un sistema de agencias de viajes tiene una parte de sus datos almacenada en una base de datos de grafos (en Neo4j) y otra parte en documentos JSON (en MongoDB).

Se garantiza que todos los datos numéricos son enteros positivos (usted no tiene que controlar esto, se garantiza que así será).

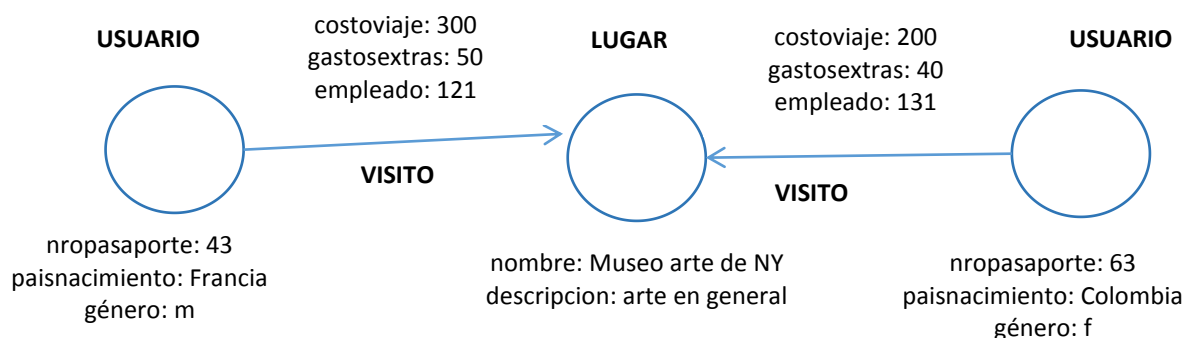
En la base de datos de grafos se tienen nodos con etiqueta **USUARIO** que tienen los siguientes atributos: **nropasaporte** (numérico), **paisnacimiento** y **género** (se garantiza que serán: **m** = masculino, **f** = femenino o **x** = otro).

También hay nodos con etiqueta **LUGAR** con los siguientes atributos: **nombre** (o sea, nombre del lugar) y **descripcion**.

Entre los nodos **USUARIO** y los nodos **LUGAR** se pueden establecer relaciones etiquetadas como **VISITO**, para indicar que un determinado usuario visitó un determinado lugar (por supuesto un usuario puede visitar muchos lugares y un lugar puede haber sido visitado por muchos usuarios; *por simplicidad suponga (se garantiza) que un mismo usuario no visita varias veces un mismo lugar*).

La relación es dirigida de **USUARIO** hacia **LUGAR**. Una relación **VISITO** tiene los siguientes atributos numéricos: **costoviaje** (es decir, cuanto pagó el usuario al empleado de la agencia por visitar un lugar), **gastosextras** (es decir, cuánto se gastó el usuario en el lugar, aparte de lo que le pagó al empleado de la agencia) y **empleado** (el código del empleado al que el usuario le pagó el **costoviaje**). **Nota:** el empleado trabaja en alguna sucursal de alguna agencia del sistema, véase la base de datos de MongoDB.

A continuación se muestra un ejemplo, por supuesto, habrá muchos más nodos en la base de datos...



Por su parte, en la base de datos MongoDB se manejan solamente **dos** colecciones. La primera colección se llama **agencias** y tiene documentos JSON con la siguiente estructura: **codigoage** (numérico, es decir, el código de una agencia de viajes) y **nombreage** (es decir, el nombre de la agencia). Por simplicidad, suponga que **no** hay códigos de agencias repetidos **ni** nombres de agencias repetidas (usted **no** tiene que controlar esto, se garantiza que durante las pruebas **no** se ingresarán datos repetidos en estos atributos).

La segunda colección se llama **sucursales** y tiene documentos JSON con la siguiente estructura: **codigosuc** (numérico, es decir, el código de una sucursal), **codagencia** (numérico, es decir, el código de la agencia a la que pertenece la sucursal) y **misempleados** (es decir, un arreglo de números que corresponde a los empleados de la sucursal).

Por simplicidad, se garantiza durante las pruebas todo lo siguiente (o sea usted **no** debe controlarlo):

- Un empleado solamente trabaja en una sucursal. Tampoco se ingresará un código de empleado repetido en un mismo arreglo.
- No habrá códigos de sucursales repetidos.
- El código de la agencia a la que pertenece la sucursal, se garantiza que existe en la colección de agencias (es como si fuese una clave foránea).

Resolver lo siguiente. Cada punto vale 25%.

1. Dado el nombre de algún lugar (por ejemplo, "Museo arte de NY"), mostrar el total gastado (**gastosextras**) por los usuarios en dicho lugar. **Mostrar los resultados discriminados por género:** es decir, cuánto gastaron los hombres, cuánto gastaron las mujeres y cuánto gastaron los usuarios de género x. **Nota:** durante las pruebas la monitora ingresará el nombre del lugar al cual desea se le calculen estos resultados. Usted le debe indicar a la monitora como ingresar dicho nombre, por ejemplo, copiándolo en una parte específica de su programa. Presente los resultados en Google Colab, usando cualquier gráfico (por ejemplo, un gráfico de código de barras).

2. Este punto es similar al anterior, igualmente dado el nombre de un lugar, pero esta vez **mostrar los resultados discriminados por género y país de nacimiento:** es decir, cuánto gastaron los hombres de Colombia, cuánto gastaron las mujeres de Colombia, cuánto gastaron las personas de género x de Colombia, cuánto gastaron los hombres de Francia, etc. en el lugar dado.

3. Obtener el **total** recaudado por cada sucursal (suma de sus **costoviajes** correspondiente a todos sus empleados). Presente los resultados en Google Colab, usando cualquier gráfico (por ejemplo, un gráfico de código de barras): mostrar el código de cada sucursal con su correspondiente total.

4. Obtener el **total** recaudado por cada agencia (suma de sus **costoviajes** correspondiente a todas sus sucursales). Presente los resultados en Google Colab,

usando cualquier gráfico (por ejemplo, un gráfico de código de barras): Mostrar el nombre de cada agencia con su correspondiente total.

Notas:

- No cambie los nombres de los atributos, etiquetas, colecciones, etc. Use los nombres indicados.
- Debe trabajar con las herramientas indicadas (Neo4j, MongoDB y Google Colab) y puede usar todas las instrucciones que desee para resolver los puntos, incluso puede usar instrucciones no vistas en clase. Si trabaja con otras herramientas, el trabajo se califica con cero.
- Trabajos copiados entre los grupos, así sea parcialmente se califican con cero.
- Este no es un trabajo orientado a limpieza ni a a calidad de datos, es decir, suponga que los datos que la monitora va a ingresar son correctos y consistentes, no se van a ingresar datos erróneos (ejemplo: no se va a ingresar una cadena de caracteres donde debería ir un número). Usted no debe hacer programas ni *triggers* para controlar que los datos estén correctos. Se garantiza que lo estarán.

Fecha de entrega: viernes 6 de septiembre hasta las 11 am, no se reciben trabajos en horario posterior. Enviar el trabajo a *fjmoreno@unal.edu.co* y a *vospinan@unal.edu.co*