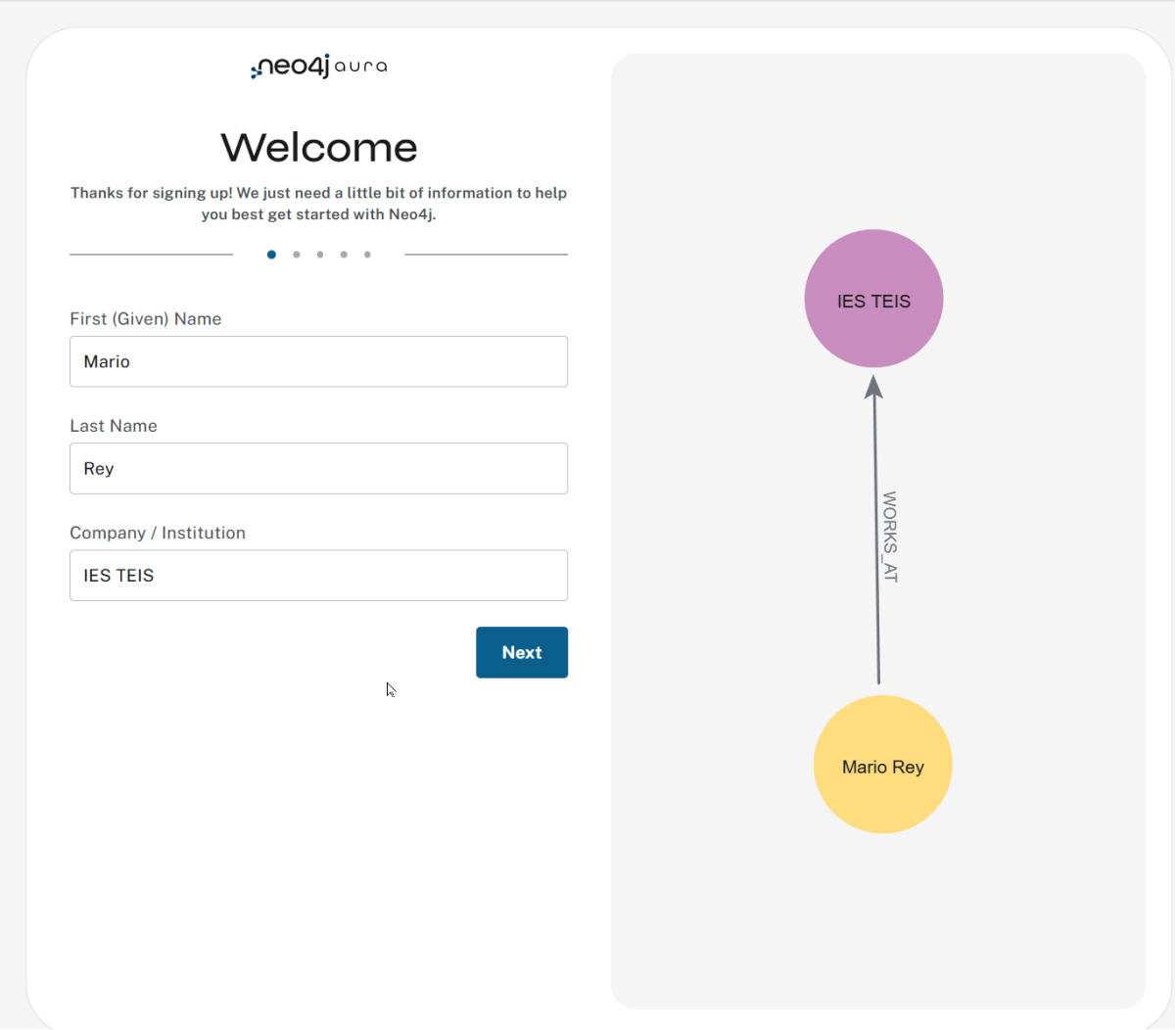
UD4. NoSQL
"Práctica Neo4J"

NOMBRE Y APELLIDOS: MARIO REY BULLIDO
DNI: 39459575Q

1.- Documenta el proceso desde la creación de una cuenta gratuita Neo4J AuraDB en https://neo4j.com/cloud/platform/aura-graph-database/ hasta tener operativa una instancia de base de datos. Captura las pantallas donde sea necesaria alguna interacción por parte del usuario y justifica las opciones que vas eligiendo.

Durante el proceso de creación de la cuenta gratuíta, la herramienta te realiza una serie de preguntas con las que va creando una base de datos relacionando cada uno de los nodos que son las respuestas a las preguntas que te realiza, las cuales establecen las relaciones entre estos nodos.

El primer nodo es el usuario como entidad, luego pregunta sobre dónde trabajo y con esta respuesta establece la relación entre estas 2 entidades, y así con el resto de preguntas.



neo4j aura

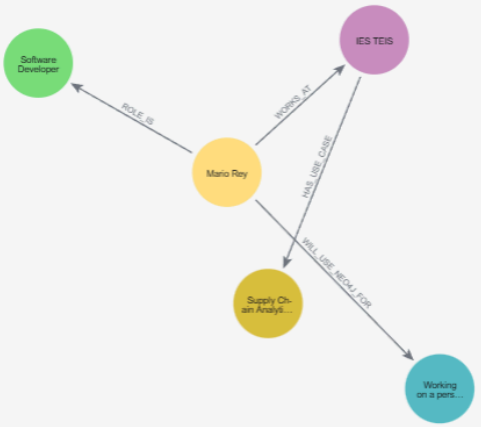
Get started with a free instance

Create instance

No credit card required

Not looking to create a free instance?

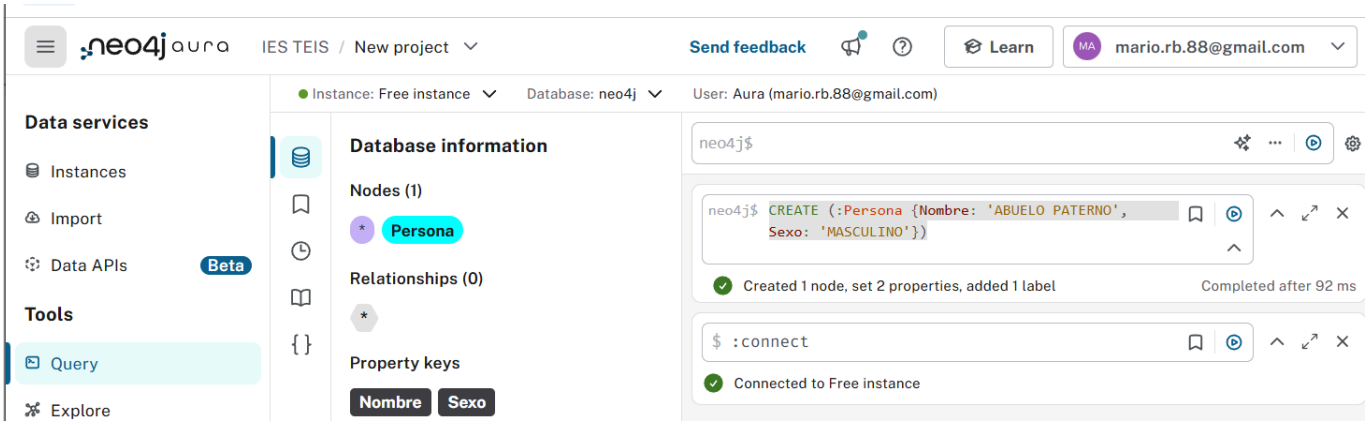
Select another instance type



2.- Crea los 12 nodos necesarios para representar la siguiente imagen.
Cada nodo debe estar etiquetado como “Persona” y tener las propiedades nombre y sexo con el valor que corresponda en cada caso.
Captura la pantalla en la que se vea el/los comandos que utilizas para crear alguno de estos nodos.
Explica con tus palabras cada parte del comando y sus parámetros.

CREATE (:Persona {Nombre: 'ABUELO PATERNO', Sexo: 'MASCULINO'})

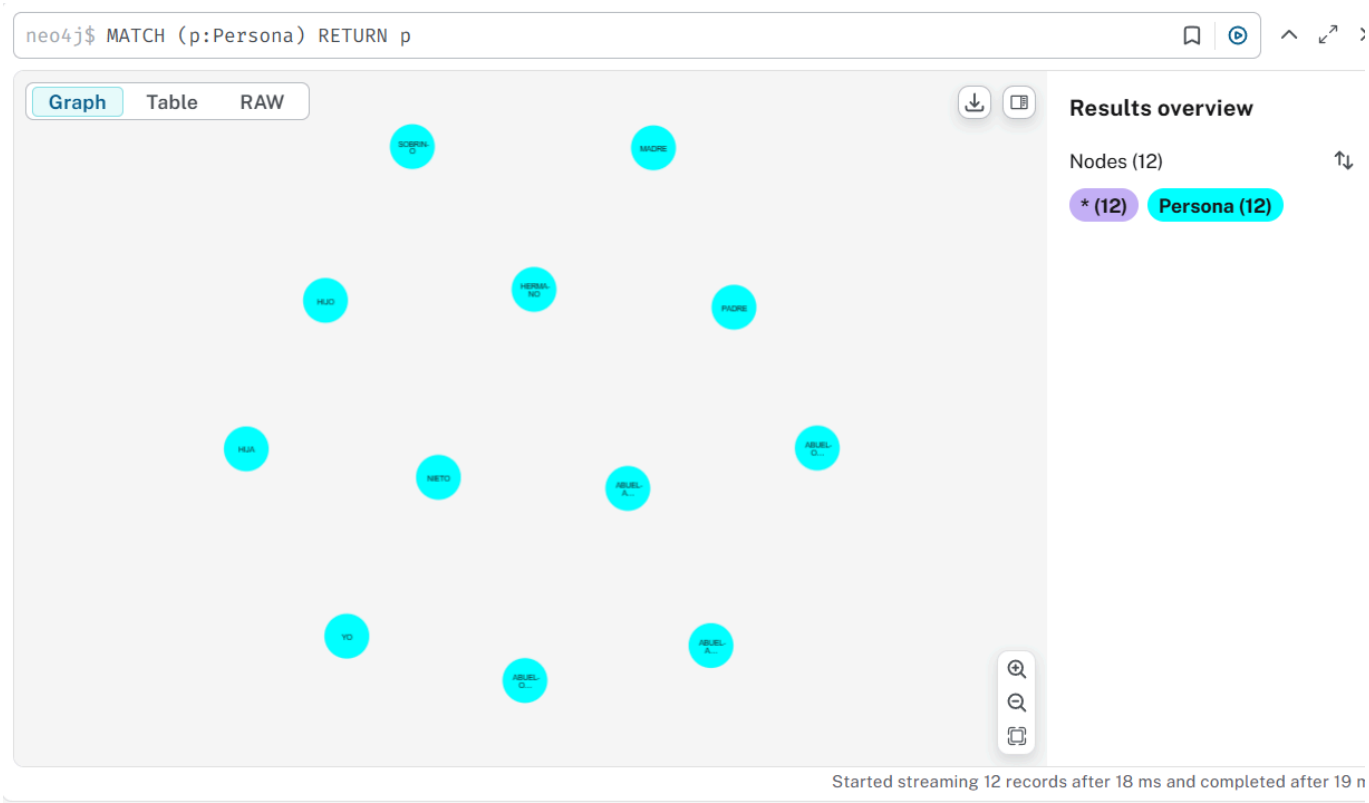
Con el comando CREATE se crea un nuevo nodo.
A continuación, con :Persona indicamos la etiqueta del nodo, que sería el tipo de entidad.
A continuación dentro de los corchetes indicamos las propiedades del nodo con parejas de clave-valor.



3.- Muestra un gráfico a través de Neo4J con todos los nodos que has creado. Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

MATCH (p:Persona) RETURN p

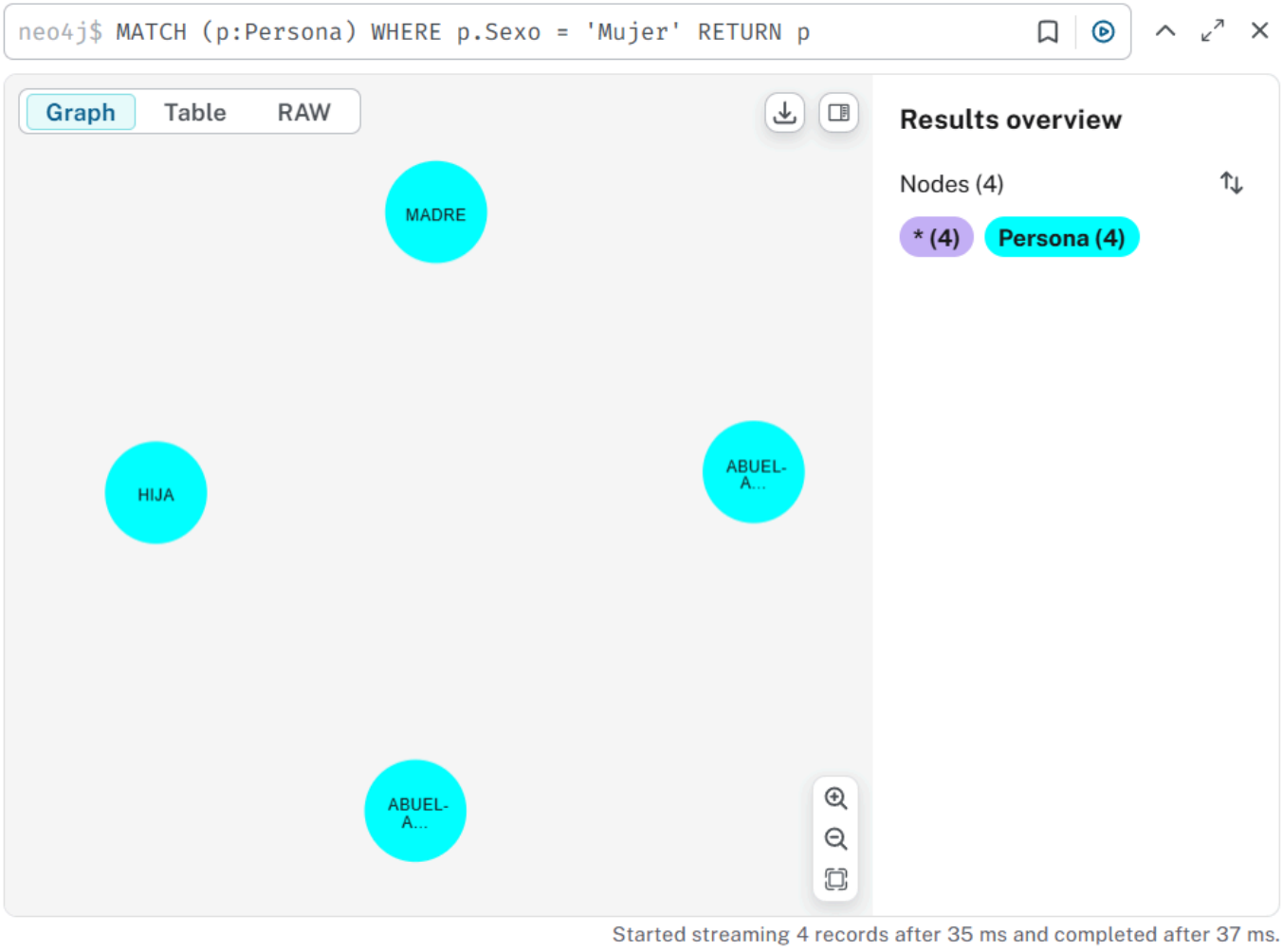
El comando MATCH se usa para indicar el patrón que queremos buscar.
Para buscar nodos se usan los paréntesis ().
Con la variables p la usaremos como alias para referencias a los nodos que estamos buscando.
Con :Persona, indicamos que buscamos los nodos con la etiqueta Persona.
Con el comando RETURN p indicamos que queremos mostrar los nodos que se filtraron con el comando MATCH



4.- Muestra un gráfico a través de Neo4J con todos los nodos de sexo “Mujer” que has creado. Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los
comandos usados y sus parámetros.

MATCH (p:Persona) WHERE p.Sexo = 'Mujer' RETURN p

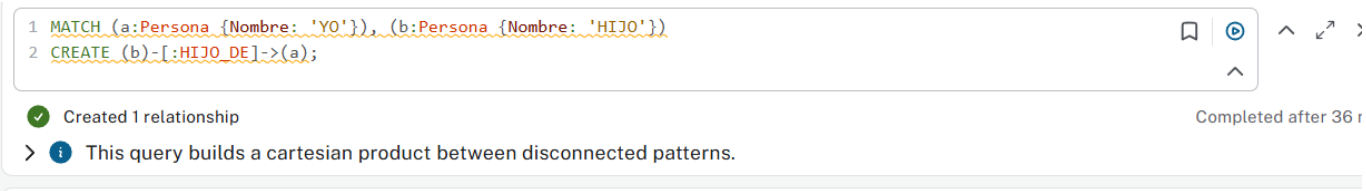
-MATCH (p:Persona) : Buscar los nodos de tipo Persona.
-WHERE p.Sexo = 'Mujer': Filtrar que los nodos tengan el valor ‘Mujer’ para la propiedad ‘Sexo’.
-RETURN p: Mostrar estos nodos filtrados.



5.- Crea las relaciones “HIJO_DE” que se ven en la imagen.
Captura la pantalla en la que se vea el/los comandos que utilizas para crear alguna de estas relaciones.
Explica con tus palabras cada parte del comando y sus parámetros.

MATCH (a:Persona {Nombre: 'YO'}), (b:Persona {Nombre: 'HIJO'})
CREATE (b)-[:HIJO_DE]->(a);

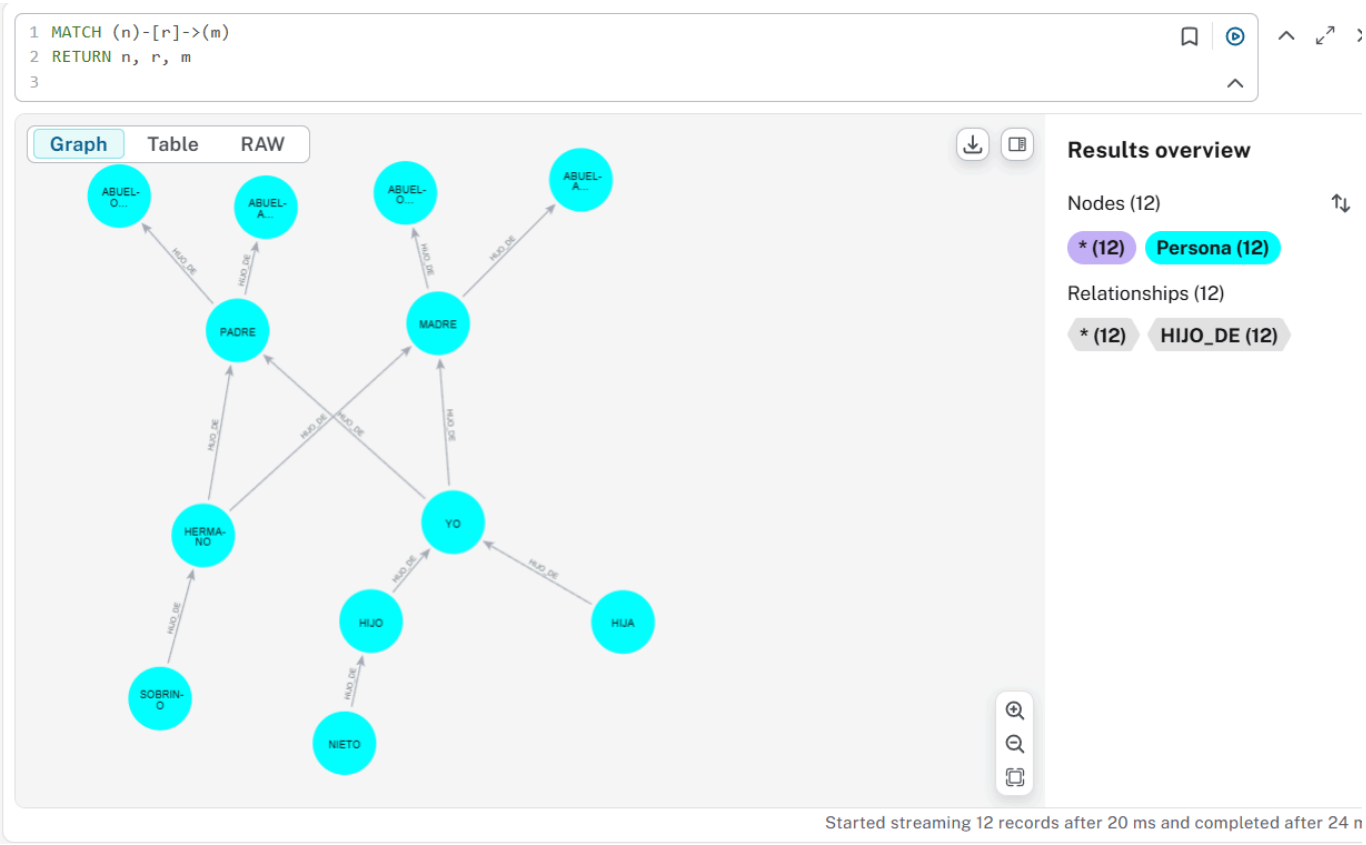
MATCH (a:Persona {Nombre: 'YO'}), (b:Persona {Nombre: 'HIJO'}): Buscamos los nodos de tipo persona que se llaman YO y HIJO y les damos los alias a y b.
CREATE (b)-[:HIJO_DE]->(a);: Creamos una relación de indicando que b(HIJO) es HIJO_DE a(YO)



6.- Muestra un gráfico a través de Neo4J en el que se vean todos los nodos y relaciones que has creado.
Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

MATCH (n)-[r]->(m)
RETURN n, r, m

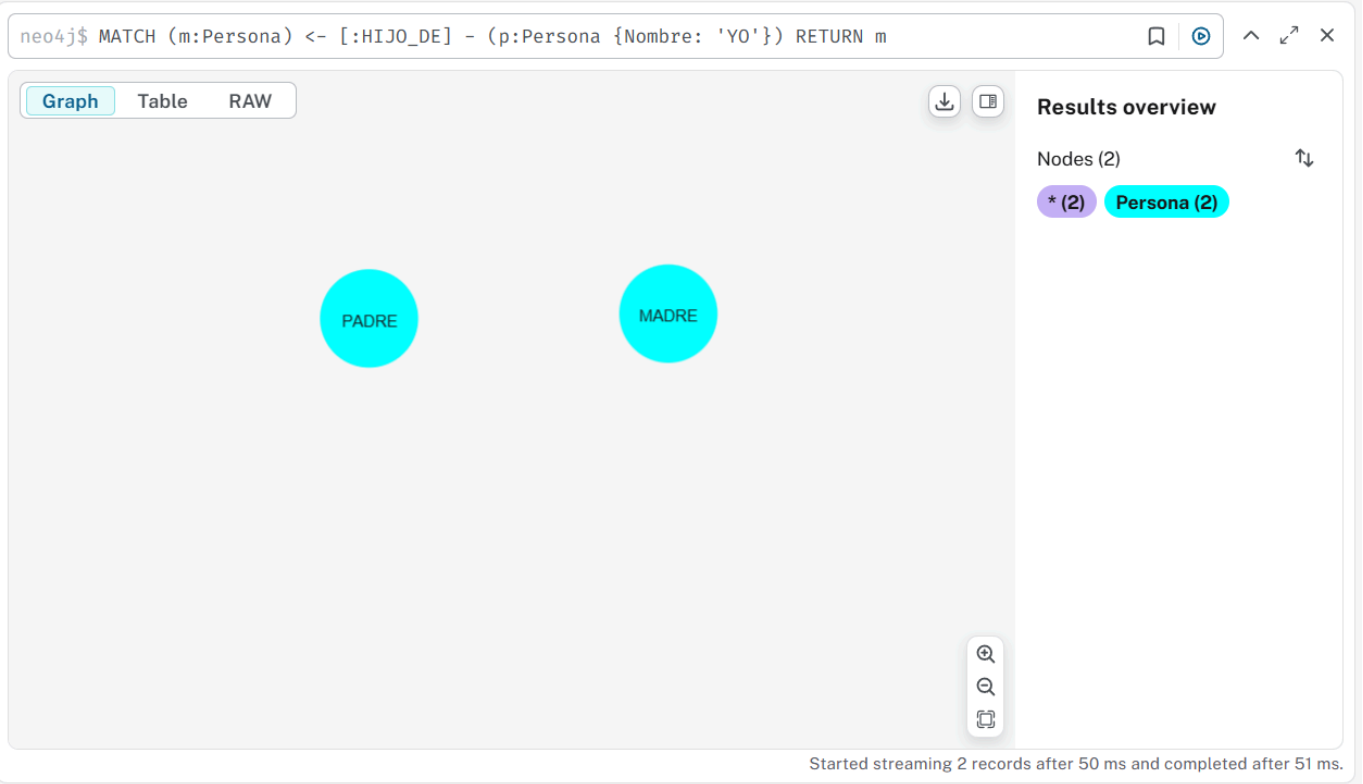
MATCH (n)-[r]->(m): Buscamos los nodos (n) que tienen una relación [r] con otros nodos (m)
RETURN n, r, m: Retornamos estos datos para visualizarlos.



7.- Muestra un gráfico en el que solo se vean los ascendentes directos de un determinado nodo, por ejemplo si buscamos para el nodo “Yo” debería aparecer los nodos “Padre” y “Madre”. Si buscamos para el nodo “Hijo” solo debo aparecer “Yo”.
Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

MATCH (m:Persona) <- [:HIJO_DE] - (p:Persona {Nombre: 'YO'})
RETURN m

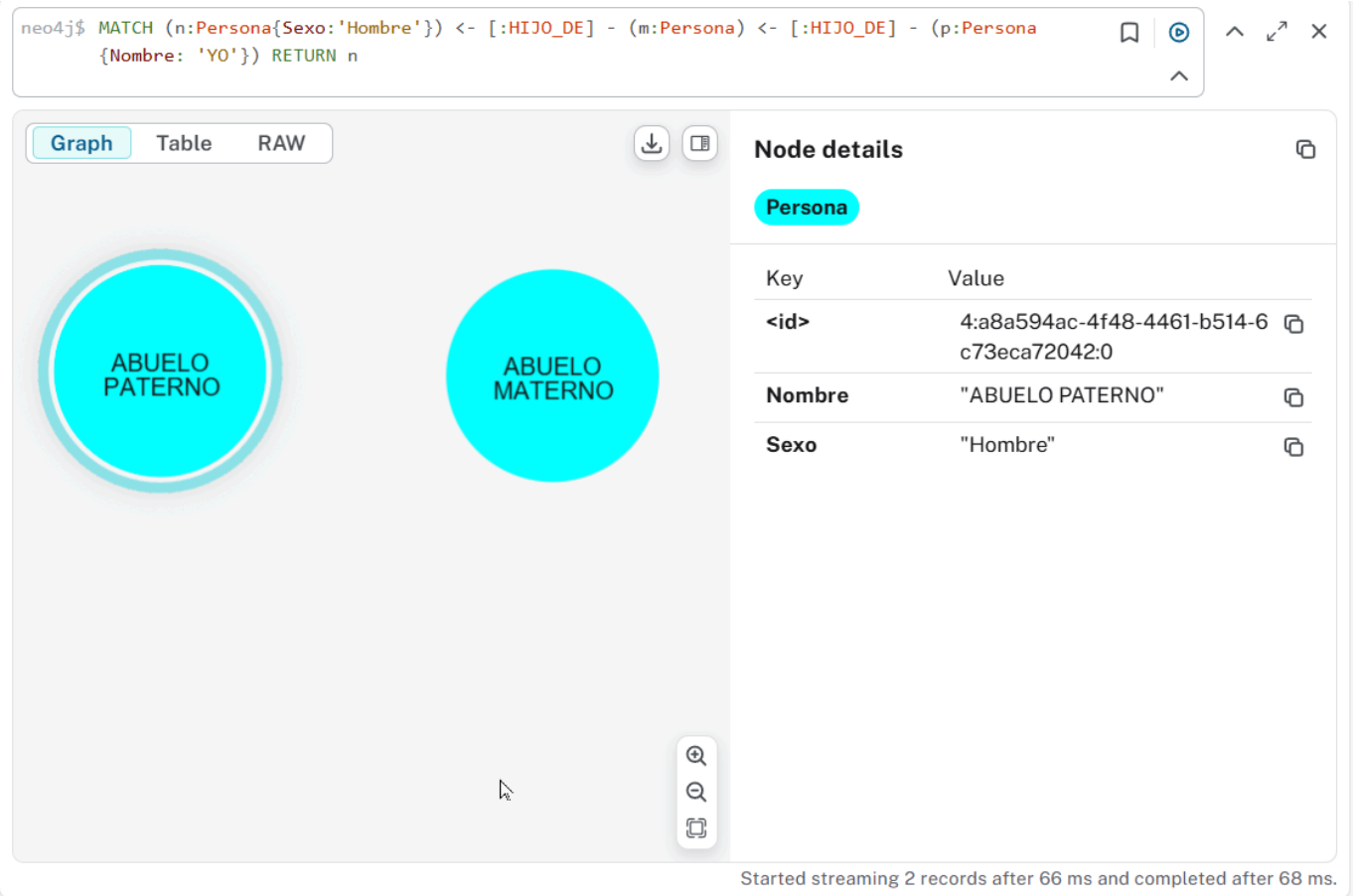
MATCH (m:Persona) <- [:HIJO_DE] - (p:Persona {Nombre: 'YO'}): Busco los nodos m de tipo Persona que tienen una relación de HIJO_DE en sentido opuesto (padres de) del nodo de tipo persona cuyo nombre es YO.



8.- Muestra un gráfico en el que se vean los segundos ascendientes directos masculinos (los abuelos) de un determinado nodo. Por ejemplo, aplicado al nodo “Yo” debería aparecer “Abuelo Materno” y Abuelo Paterno”. Aplicado al nodo “Hijo” solo debería aparecer “Padre”. Aplicado a “Padre” no debería aparecer
nada.
Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

MATCH (n:Persona{Sexo:'Hombre'}) <- [:HIJO_DE] - (m:Persona) <- [:HIJO_DE] - (p:Persona {Nombre: 'YO'}) RETURN n

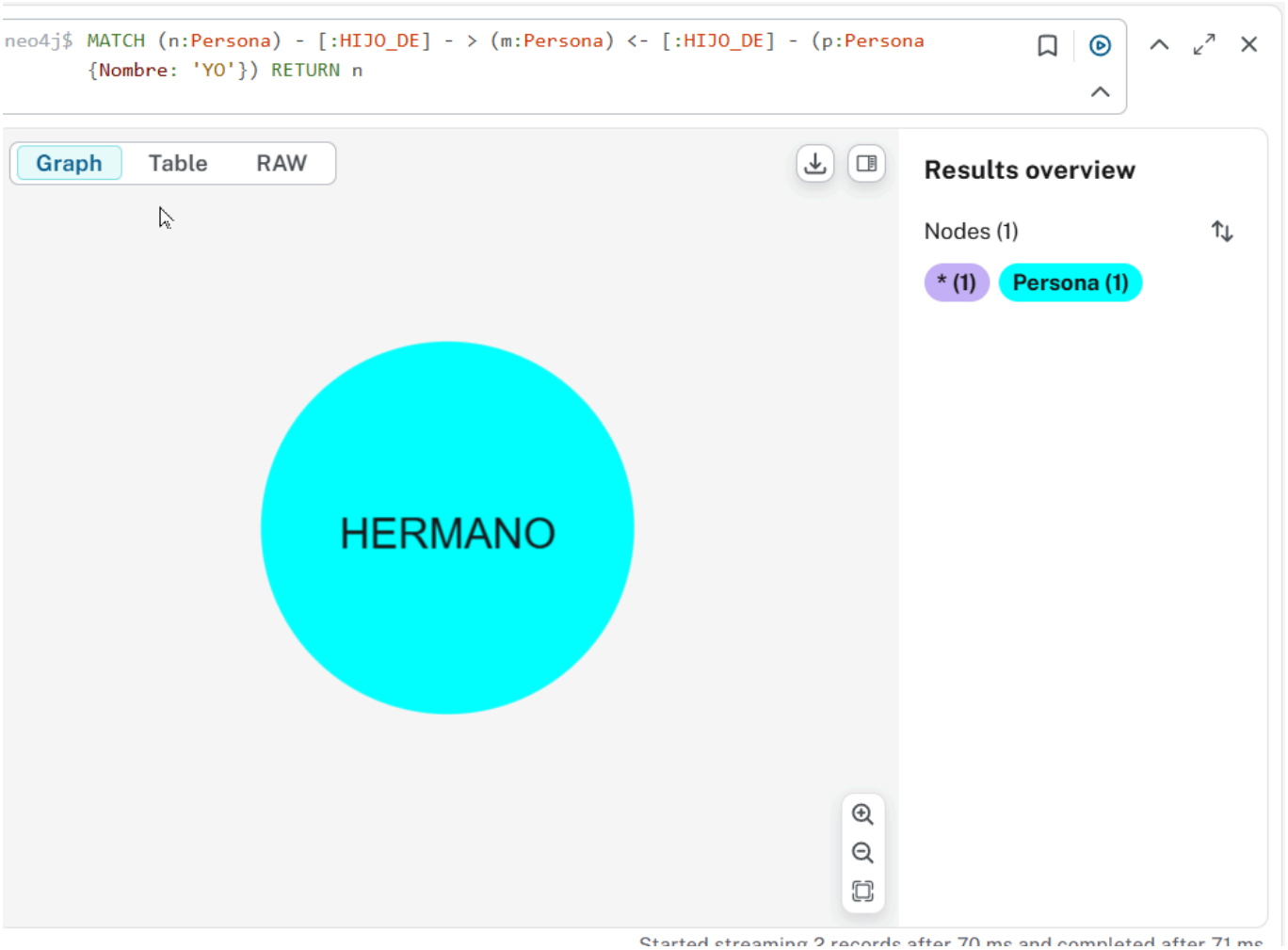
Busco nos nodos persona de sexo Hombre (alias n) que tienen una relación inversa de HIJO_DE con otros nodos persona (alias m) que tienen otra relación inversa de HIJO_DE con el nodo persona de nombre YO (alias p).



8.- Muestra un gráfico en el que se vean los hermanos de un nodo (los que comparten algún ascendiente directo). Por ejemplo, aplicado al nodo “Yo” debería aparecer el nodo “Hermano”, aplicado a “Hija” debería aparecer “Hijo”, aplicado a “Nieto” no debería aparecer nada y aplicado a “Madre” tampoco debería aparecer nada.
Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

MATCH (n:Persona) - [:HIJO_DE] - > (m:Persona) <- [:HIJO_DE] - (p:Persona {Nombre: 'YO'}) RETURN n

Busco un nodo persona (alias n) que tiene tiene una relación HIJO_DE con un nodo persona (alias m) que tiene una relación inversa de HIJO_DE con un nodo persona (alias p) cuyo nombre es YO.



9.- Muestra un gráfico en el que se vean los tíos de un nodo. Por ejemplo, aplicado al nodo "Sobrino"
debería aparecer yo. Aplicado a "Hijo" debería aparecer "Hermano" y aplicado a "Yo" no debería aparecer nada.
Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

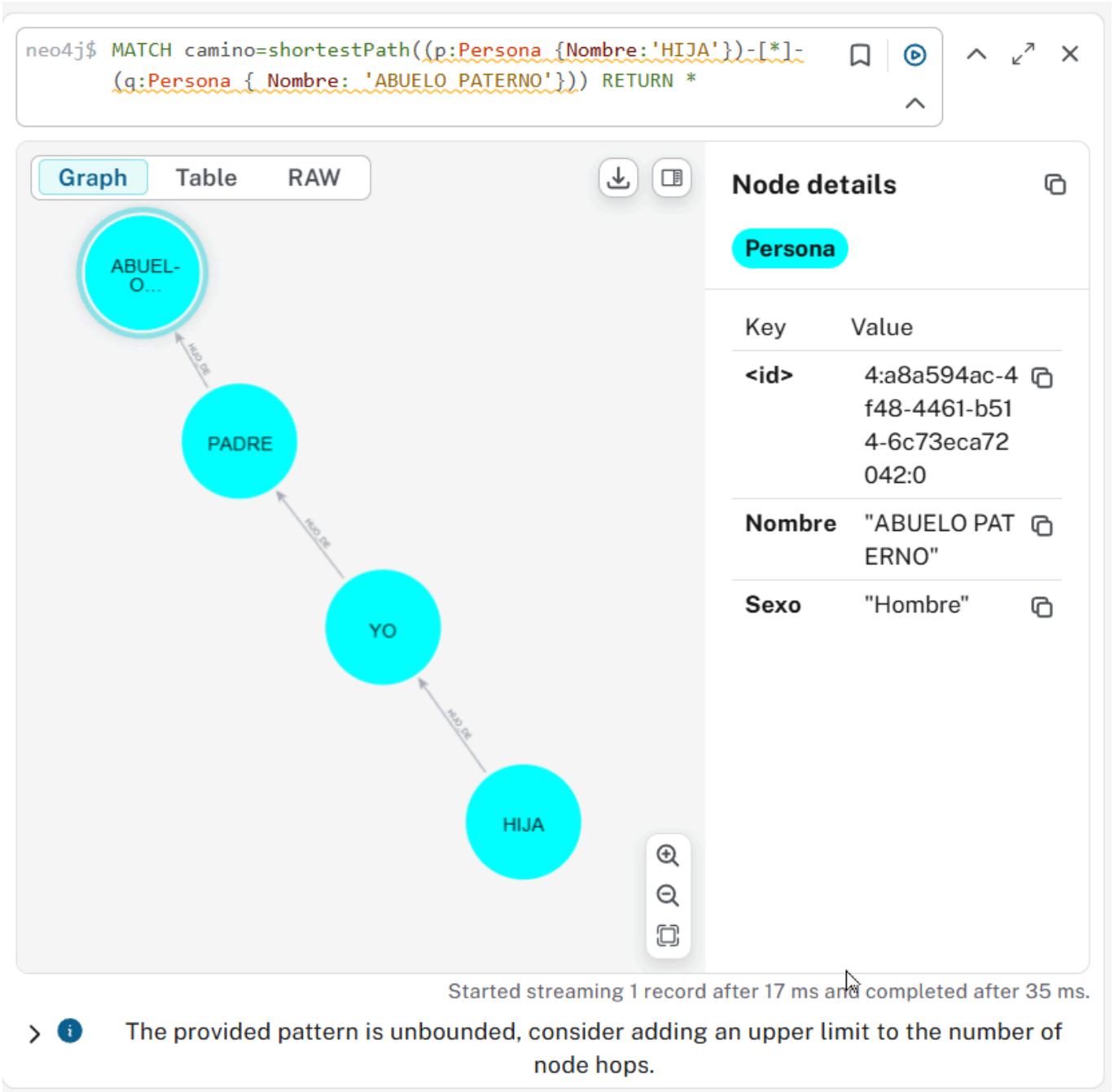
MATCH (n:Persona) - [:HIJO_DE] - > (m:Persona) <- [:HIJO_DE] - (o:Persona) <- [:HIJO_DE] - (p:Persona {Nombre: 'HIJA'}) RETURN n

Busco un nodo persona (alias n) que tiene tiene una relación HIJO_DE con un nodo persona (alias m) que tiene una relación inversa de HIJO_DE con un nodo persona (alias o) que tiene una relación inversa de HIJO_DE con un nodo persona (alias p) cuyo Nombre es HIJA.

10.-Muestra un gráfico en el que se vea la ruta más corta entre dos nodos cualesquiera. Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

MATCH camino=shortestPath((p:Persona {Nombre:'HIJA'})-[*]-(q:Persona { Nombre: 'ABUELO PATERNO'})) RETURN *

Uso la variable “camino” para incluir los nodos y relaciones que obtengo de usar la función shortestPath (que busca el camino más corto entre nodos) a la que le paso como parámetros el nodo persona (alias p) cuyo nombre es HIJA, el nodo persona (alias q) cuyo nombre es ABUELO y le indico que puede haber cualquier relación entre ellos mediante -[*]-.



11.-Muestra un gráfico en el que se vea el primer nodo en común entre dos nodos cualesquiera. Captura la pantalla donde se vea el comando que utilizas y el gráfico resultante. Explica con tus palabras los comandos usados y sus parámetros.

```
MATCH path1 = (a:Persona {Nombre: 'HIJO'})-[:HIJO_DE*]->(comun)
MATCH path2 = (b:Persona {Nombre: 'SOBRINO'})-[:HIJO_DE*]->(comun)
RETURN comun, length(path1) + length(path2) AS distanciaTotal
ORDER BY distanciaTotal ASC
LIMIT 1
```

Calcula el camino desde HIJO y SOBRINO hasta los nodos en común.
Suma las distancias de cada nodo con cada nodo en común y ordena los nodos comunes y sus distancias para poder ordenar de manera ascendente y quedarnos con el primero.