

## Módulo 6. TECNOLOGÍAS DE ALMACENAMIENTO ESCALABLE

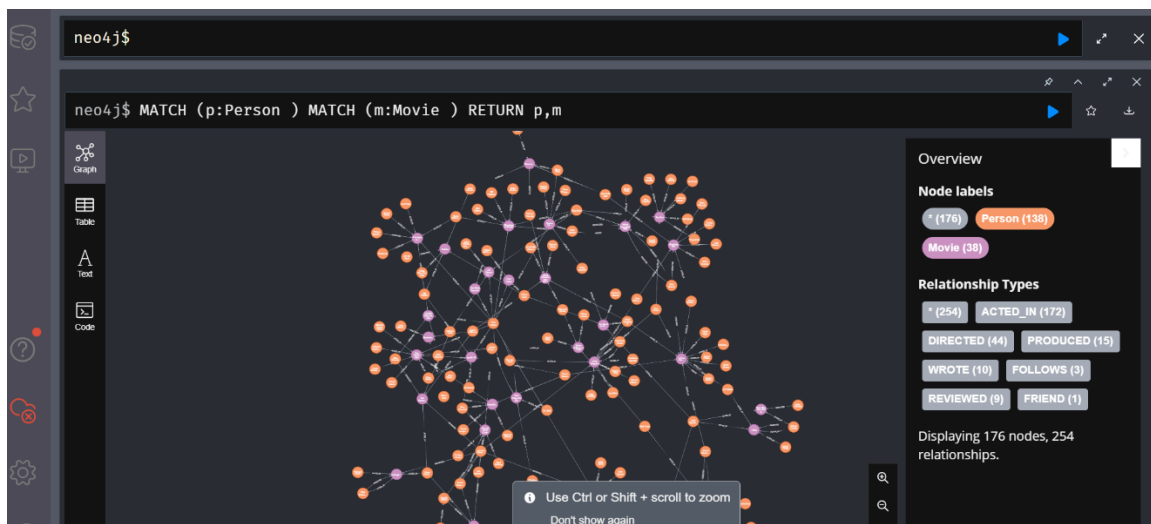
### ACTIVIDAD DE MONGODB Y NEO4J

#### Ejercicios Neo4J:

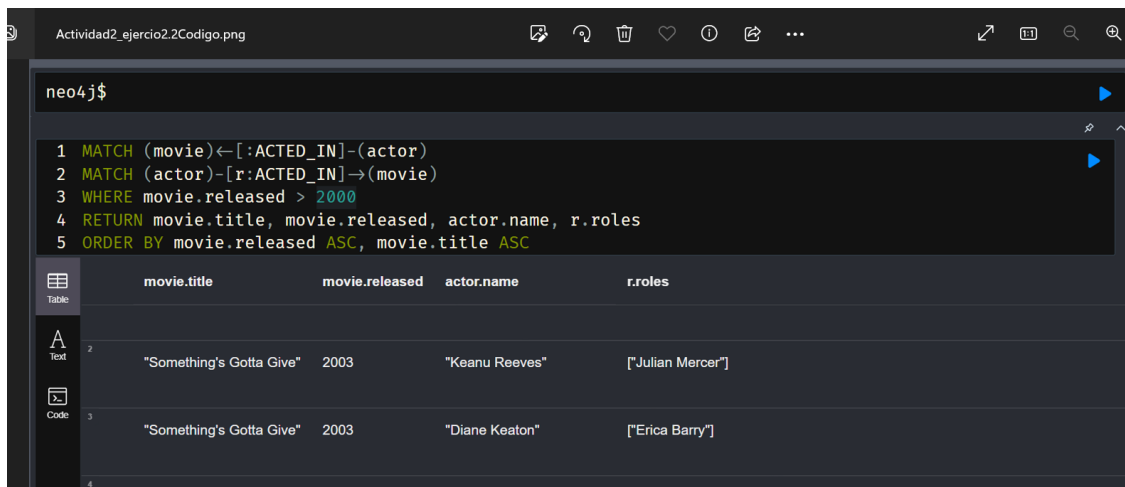
1. Obtener el total de Personas y de Películas de la base de datos

```
neo4j$ MATCH (p:Person ) MATCH (m:Movie ) RETURN p,m
```

Con Match se solicita que encuentre las etiquetas de nodo, a las cuales se les asigna parámetros de llamada mas breves p y m. Para “imprimir” o devolver en el resultado todas las películas y personas.



2. Obtener, para las películas estrenadas después del año 2000, su título, el año en que se estrenó, el nombre de los actores que han participado en la misma, y su rol (es decir, el nombre de su personaje en la película). Los resultados deben estar ordenados por año de estreno y, dentro de cada año, alfabéticamente por el nombre de la película.



The screenshot shows a Neo4j Cypher query interface. The query is as follows:

```
1 MATCH (movie)←[:ACTED_IN]-(actor)
2 MATCH (actor)-[:ACTED_IN]→(movie)
3 WHERE movie.released > 2000
4 RETURN movie.title, movie.released, actor.name, r.roles
5 ORDER BY movie.released ASC, movie.title ASC
```

The results are displayed in a table with the following columns: movie.title, movie.released, actor.name, and r.roles. The table contains two rows of data.

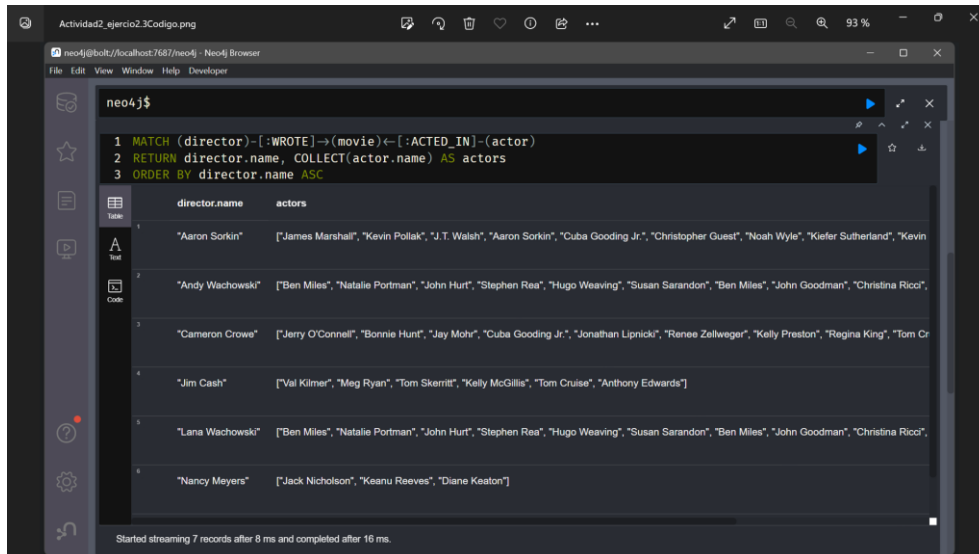
	movie.title	movie.released	actor.name	r.roles
2	"Something's Gotta Give"	2003	"Keanu Reeves"	["Julian Mercer"]
3	"Something's Gotta Give"	2003	"Diane Keaton"	["Erica Barry"]

Se hace MATCH para que seleccione las películas en las que han actuado actores por un lado y los actores que han actuado en películas. Con función WHERE indicamos que restrinja la selección a las películas estrenadas con posterioridad al 2000 con el operador >.

A continuación, con RETURN solicitamos que muestre (bajo la condición anterior) el título de película, el año de estreno, el nombre del actor y el rol de los actores.

Para terminar, el output se ordena de manera ascendente por fecha de estreno y por título de película.

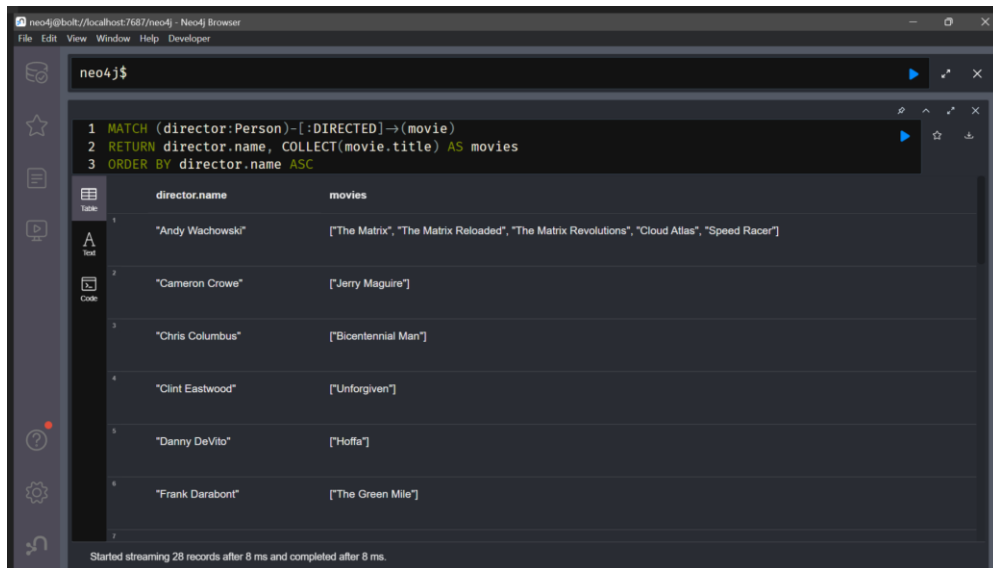
3. Obtener, para cada guionista, su nombre y el nombre de los actores que han trabajado en sus películas. Se espera que haya una fila por cada guionista, por lo tanto, cada fila resultado deberá contener el nombre del guionista y una lista con el nombre de los actores asociados.



Primero debemos seleccionar MATCH los directores que han escrito una película, en la cual han actuado los actores. (Nota: director==guionista, lapsus al escribir el código).

Llamamos entonces al nombre de los guionistas y para reunir todos los actores con los que ha trabajado, se indica COLLECT, creando así una lista que agrupa los nombres de los actores por cada guionista. Finalmente se ordena por nombre de guionista.

4. Obtener, para cada director, el nombre de las películas que ha dirigido.



The screenshot shows the Neo4j Browser interface. The Cypher query is as follows:

```
1 MATCH (director:Person)-[:DIRECTED]->(movie)
2 RETURN director.name, COLLECT(movie.title) AS movies
3 ORDER BY director.name ASC
```

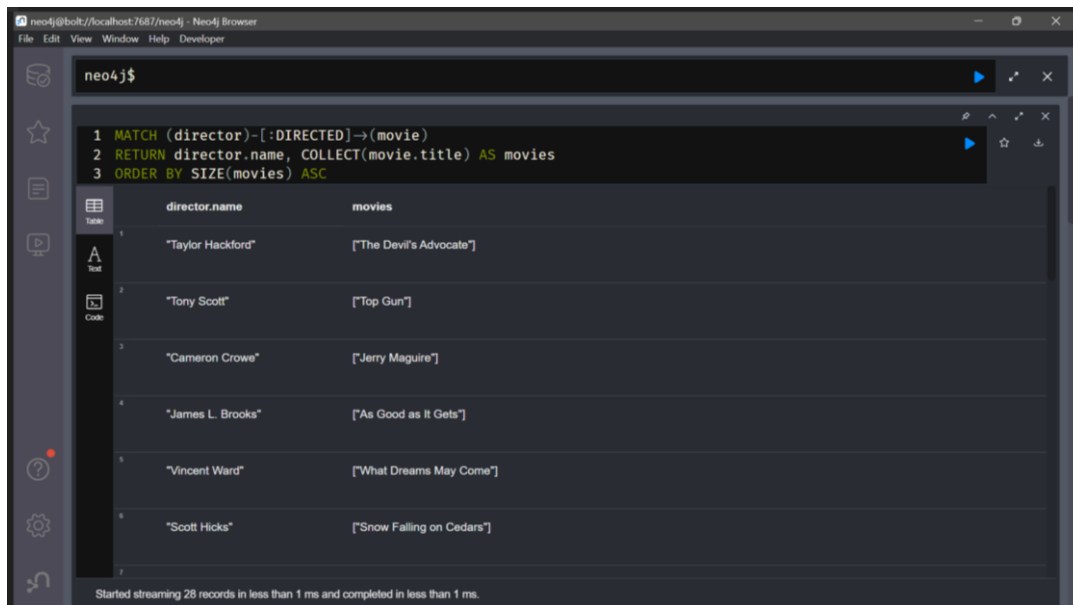
The results are displayed in a table with two columns: **director.name** and **movies**. The table contains 6 rows of data, ordered by director name.

director.name	movies
"Andy Wachowski"	["The Matrix", "The Matrix Reloaded", "The Matrix Revolutions", "Cloud Atlas", "Speed Racer"]
"Cameron Crowe"	["Jerry Maguire"]
"Chris Columbus"	["Bicentennial Man"]
"Clint Eastwood"	["Unforgiven"]
"Danny DeVito"	["Hoffa"]
"Frank Darabont"	["The Green Mile"]

At the bottom of the interface, it states: "Started streaming 28 records after 8 ms and completed after 8 ms."

Seleccionamos las personas (directores) que han dirigido películas y llamamos a los nombres de los directores, agrupando en lista (COLLECT) los títulos de películas que han dirigido. Ordenamos por nombre de director de manera ascendente.

5. Ordenad el resultado anterior de forma ascendente en función del número de películas. Es decir, los directores que han dirigido menos películas primero. En este caso, es necesario presentar, para cada director, una lista con las películas con las que ha trabajado.



The screenshot shows the Neo4j Browser interface. At the top, the query editor contains the following Cypher query:

```
1 MATCH (director)-[:DIRECTED]->(movie)
2 RETURN director.name, COLLECT(movie.title) AS movies
3 ORDER BY SIZE(movies) ASC
```

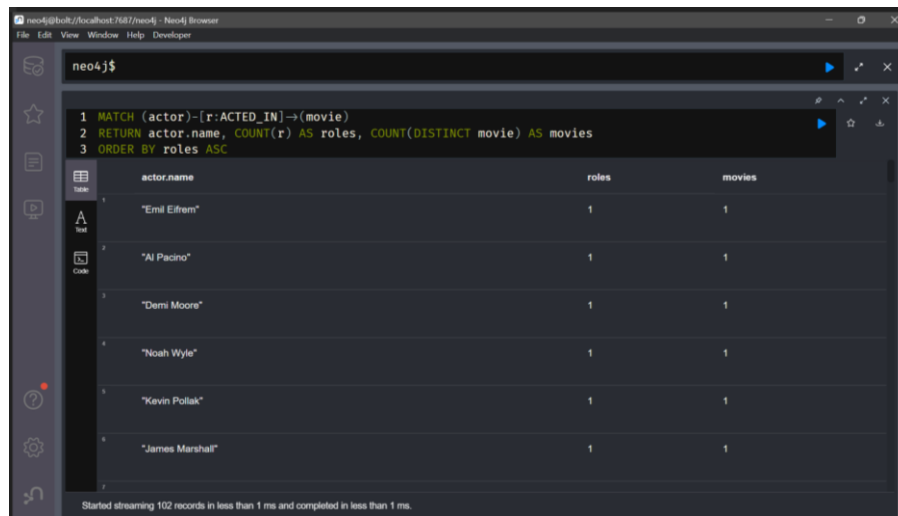
Below the query editor, the results are displayed in a table with two columns: 'director.name' and 'movies'. The results are ordered by the number of movies directed, from least to most.

	director.name	movies
1	"Taylor Hackford"	["The Devil's Advocate"]
2	"Tony Scott"	["Top Gun"]
3	"Cameron Crowe"	["Jerry Maguire"]
4	"James L. Brooks"	["As Good as It Gets"]
5	"Vincent Ward"	["What Dreams May Come"]
6	"Scott Hicks"	["Snow Falling on Cedars"]
7		

At the bottom of the interface, a status message reads: "Started streaming 28 records in less than 1 ms and completed in less than 1 ms."

Seleccionamos MATCH las personas (directores) que han dirigido películas. Llamamos a los nombres de las personas (directores) y agrupamos en lista (COLLECT) el nombre de las películas que han dirigido. Ordenamos por cantidad de nombres de películas dentro de la lista (COLLECT), de manera ascendente. Si se cambia a descendente, veremos el director con mas de un título en la lista.

6. Obtener el número de papeles interpretados por cada actor y el número de películas donde ha actuado. Tened en cuenta que el número de papeles y el número de películas de un actor pueden ser distintos, ya que un actor puede realizar distintos papeles en una película. Mostrar los resultados ordenados ascendente por número de papeles.



The screenshot shows the Neo4j Browser interface. At the top, the command line shows 'neo4j\$'. Below it, a Cypher query is entered in a text editor:

```
1 MATCH (actor)-[r:ACTED_IN]->(movie)
2 RETURN actor.name, COUNT(r) AS roles, COUNT(DISTINCT movie) AS movies
3 ORDER BY roles ASC
```

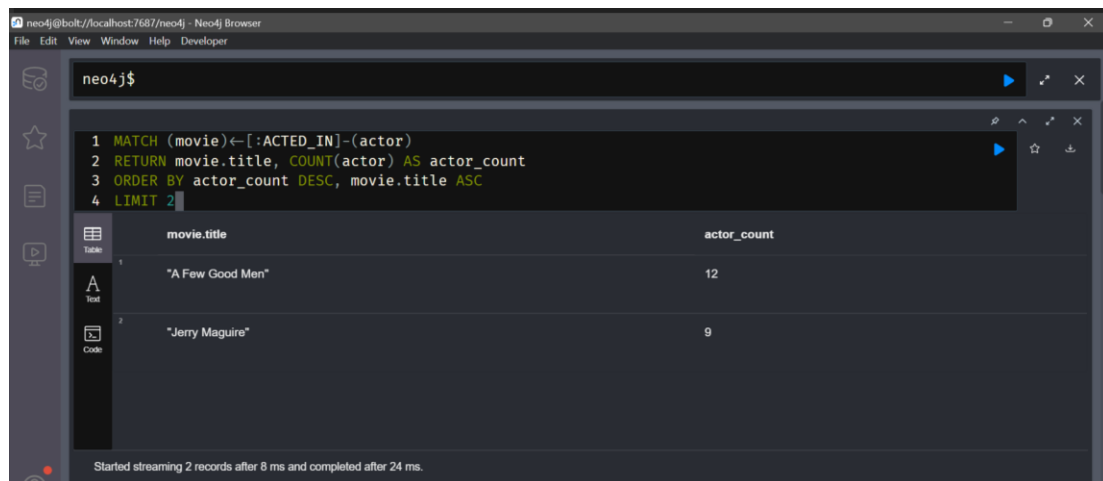
The results are displayed in a table with three columns: 'actor.name', 'roles', and 'movies'. The table contains six rows of data, ordered by the number of roles (which is 1 for all actors in this result set).

	actor.name	roles	movies
1	"Emil Eifrem"	1	1
2	"Al Pacino"	1	1
3	"Demi Moore"	1	1
4	"Noah Wyle"	1	1
5	"Kevin Pollak"	1	1
6	"James Marshall"	1	1

At the bottom of the interface, a status message reads: 'Started streaming 102 records in less than 1 ms and completed in less than 1 ms.'

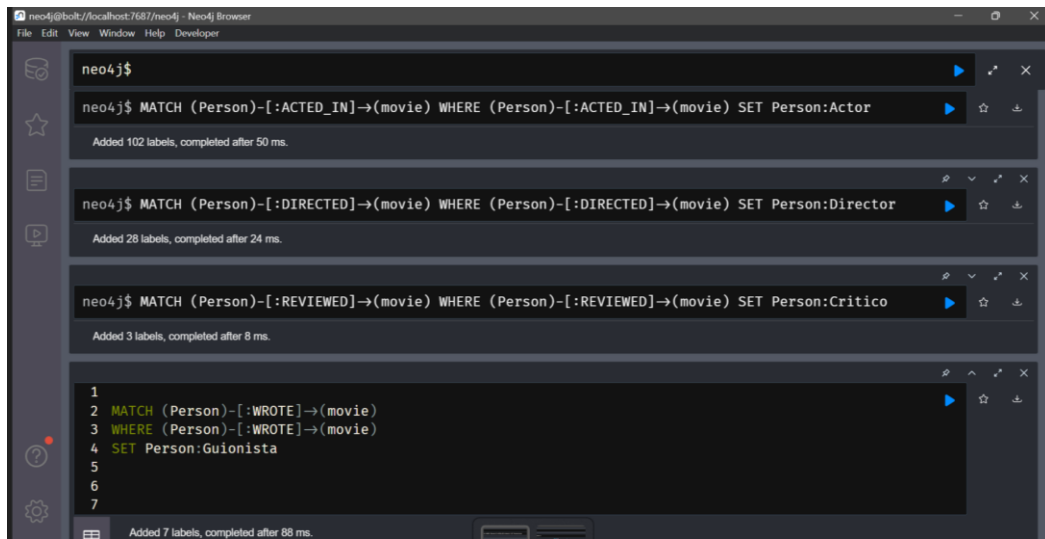
MATCH de las personas (actores) que han actuado en al menos una película. La relación ACTED\_IN se le asigna identificación "r" para facilitar operar posteriormente con dicha la selección obtenida basada en esta relación. RETURN el nombre de los actores, y cuenta del número de relaciones r que han participado (es decir, el número de papeles(roles) que han interpretado) y el número de películas distintas en las que han actuado. Se ordena por roles.

## 7. ¿Cuáles son las 2 películas con más actores?



MATCH para encontrar todas las películas que han tenido al menos un actor que ha actuado en ellas. La cláusula RETURN devolverá el título de la película y el número de actores que han actuado. ORDER BY ordena los resultados en orden descendente en función del número de actores que han actuado en cada película y en orden ascendente el nombre de la película. La cláusula LIMIT se utiliza para limitar el número de resultados a dos, de modo que solo se devuelvan las dos películas con más actores.

8. Etiquetar con las etiquetas Actor, Director, Guionista y Critico a los actores, directores, guionistas y críticos de la base de datos.

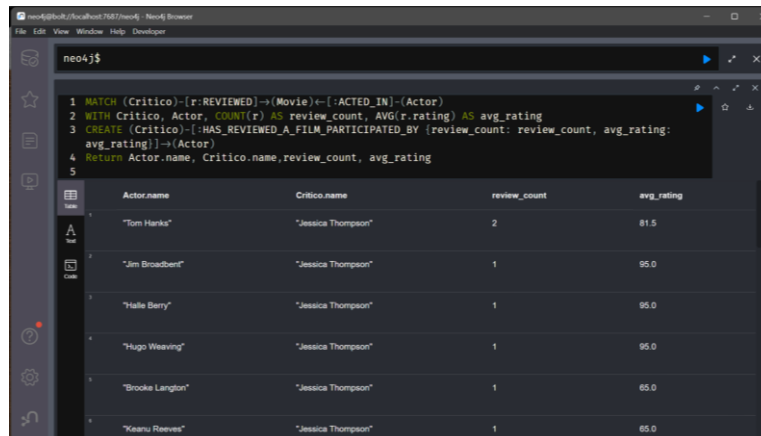


```
neo4j$  
neo4j$ MATCH (Person)-[:ACTED_IN]-(movie) WHERE (Person)-[:ACTED_IN]-(movie) SET Person:Actor  
Added 102 labels, completed after 50 ms.  
  
neo4j$ MATCH (Person)-[:DIRECTED]-(movie) WHERE (Person)-[:DIRECTED]-(movie) SET Person:Director  
Added 28 labels, completed after 24 ms.  
  
neo4j$ MATCH (Person)-[:REVIEWED]-(movie) WHERE (Person)-[:REVIEWED]-(movie) SET Person:Critico  
Added 3 labels, completed after 8 ms.  
  
1  
2 MATCH (Person)-[:WROTE]-(movie)  
3 WHERE (Person)-[:WROTE]-(movie)  
4 SET Person:Guionista  
5  
6  
7  
Added 7 labels, completed after 88 ms.
```

MATCH para encontrar todas las personas que tienen una propiedad ACTED\_IN, DIRECTED, WROTE o REVIEWED, respectivamente. La cláusula WHERE se utiliza para filtrar los resultados e incluir a las personas que cumplen con la relación indicada. Finalmente, la cláusula SET se utiliza para agregar la etiqueta Actor, Director, Guionista o Critico a cada persona que cumpla con los criterios de la consulta.



9. [...]crear una relación de tipo “HAS\_REVIEWED\_A\_FILM\_PARTICIPATED\_BY” que relacione los críticos con los actores que han participado en las películas donde el crítico ha escrito alguna crítica. La relación debe contener el número de veces donde han coincidido y la media de ratings que ha dado a sus películas. [...]

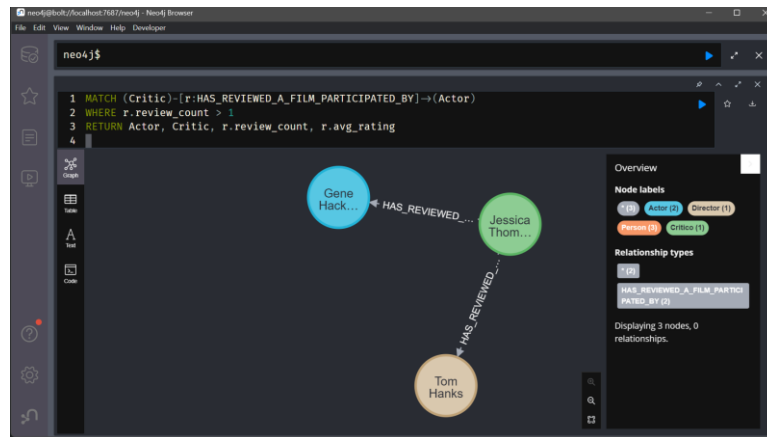


The screenshot shows the Neo4j Browser interface. The top panel displays a Cypher query with the following steps: 1. MATCH (Critico)-[:REVIEWED]->(Movie)-[:ACTED\_IN]->(Actor), 2. WITH Critico, Actor, COUNT(r) AS review\_count, AVG(r.rating) AS avg\_rating, 3. CREATE (Critico)-[:HAS\_REVIEWED\_A\_FILM\_PARTICIPATED\_BY]->(Actor) with properties review\_count and avg\_rating, 4. RETURN Actor.name, Critico.name, review\_count, avg\_rating. The bottom panel shows the results of the query as a table with four columns: Actor.name, Critico.name, review\_count, and avg\_rating. The results list seven actors, all associated with 'Jessica Thompson' as the critic, with review counts ranging from 1 to 2 and average ratings from 65.0 to 81.5.

Actor.name	Critico.name	review_count	avg_rating
"Tom Hanks"	"Jessica Thompson"	2	81.5
"Jim Broadbent"	"Jessica Thompson"	1	95.0
"Halle Berry"	"Jessica Thompson"	1	95.0
"Hugo Weaving"	"Jessica Thompson"	1	95.0
"Brooke Langton"	"Jessica Thompson"	1	65.0
"Keanu Reeves"	"Jessica Thompson"	1	65.0

MATCH para encontrar a todos los críticos que han escrito alguna crítica “r” para una película, y también encuentra a todos los actores a que han actuado en esa película. La cláusula WITH se utiliza para pasar el resultado de la búsqueda a la siguiente etapa de la consulta, y para calcular el número de veces que el crítico y el actor han coincidido en una película (review\_count) y la media de ratings que el crítico ha dado a las películas en las que el actor ha actuado (avg\_rating). Por último, se utiliza la cláusula CREATE para crear una nueva relación de tipo “HAS\_REVIEWED\_A\_FILM\_PARTICIPATED\_BY”

10. Una vez cerrado el punto anterior, realizad una consulta que permita obtener los críticos y actores que han coincidido más de una vez, es decir donde el crítico haya revisado más de una de las películas del autor. Devolved el nombre del actor, el nombre del guionista, el número de veces que han coincidido y la media de ratings.



MATCH encuentra todos los críticos que han escrito críticas para películas en las que ha actuado un actor, y luego WHERE para filtrar los resultados y solo incluir aquellos en los que el crítico y el actor han coincidido más de una vez (review\_count sea mayor que 1). La cláusula RETURN el nombre del actor, el nombre del crítico, el número de veces que han coincidido (review\_count) y la media de ratings (avg\_rating).