



Ingeniería en Sistemas Computacionales

Bases de Datos No Relacionales

Práctica en Clase: Índices en Neo4j

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 24 de abril de 2023

A manera de práctica breve acerca de la creación, uso y análisis de índices en Neo4j, se propone la elaboración de una serie de actividades que demuestren el dominio de lo anterior descrito.

Creación de índices

```
// Range
CREATE RANGE INDEX tipo_posicion_idx IF NOT EXISTS
FOR (j:Jugador)
ON (j.estatura, j.peso, j.numero);

// Text
CREATE TEXT INDEX municipalidad_idx IF NOT EXISTS
FOR ()-[r:CASA_DE]-()
ON (r.municipalidad);

// Full-text
CREATE FULLTEXT INDEX player_name_idx IF NOT EXISTS
FOR (n:Jugador|Equipo|Conferencia|Estado)
ON EACH [n.nombre];
```

Demostración de índices

```
// tipo_posicion_idx
PROFILE
MATCH (j:Jugador)
WHERE j.estatura > 1.8
AND j.peso < 110
AND j.numero IN range(1, 15)
RETURN j;
```

neo4j\$ PROFILE MATCH (j:Jugador) WHERE j.estatura > 1.8 AND j.peso < 110 AND j.numero IN range(1, 15) RETURN j;

The image displays the Neo4j Cypher Studio interface. On the left, a sidebar contains icons for Graph, Table, Text, Plan, and Code. The main area shows a query plan for the executed Cypher query. The plan consists of three main stages: **NodeIndexSeek@neo4j**, **Filter@root**, and **ProduceResults@neo4j**. The **NodeIndexSeek** stage is highlighted with a blue arrow, indicating it is the current step. Below the plan, a **Result** tab shows the output of the query. In the bottom left, a graph visualization shows three orange circular nodes. To the right of the graph, an **Overview** panel displays **Node labels** as *** (3)** and **Jugador (3)**, and states **Displaying 3 nodes, 0 relationships.**

```
//municipalidad_idx
```

```
PROFILE
```

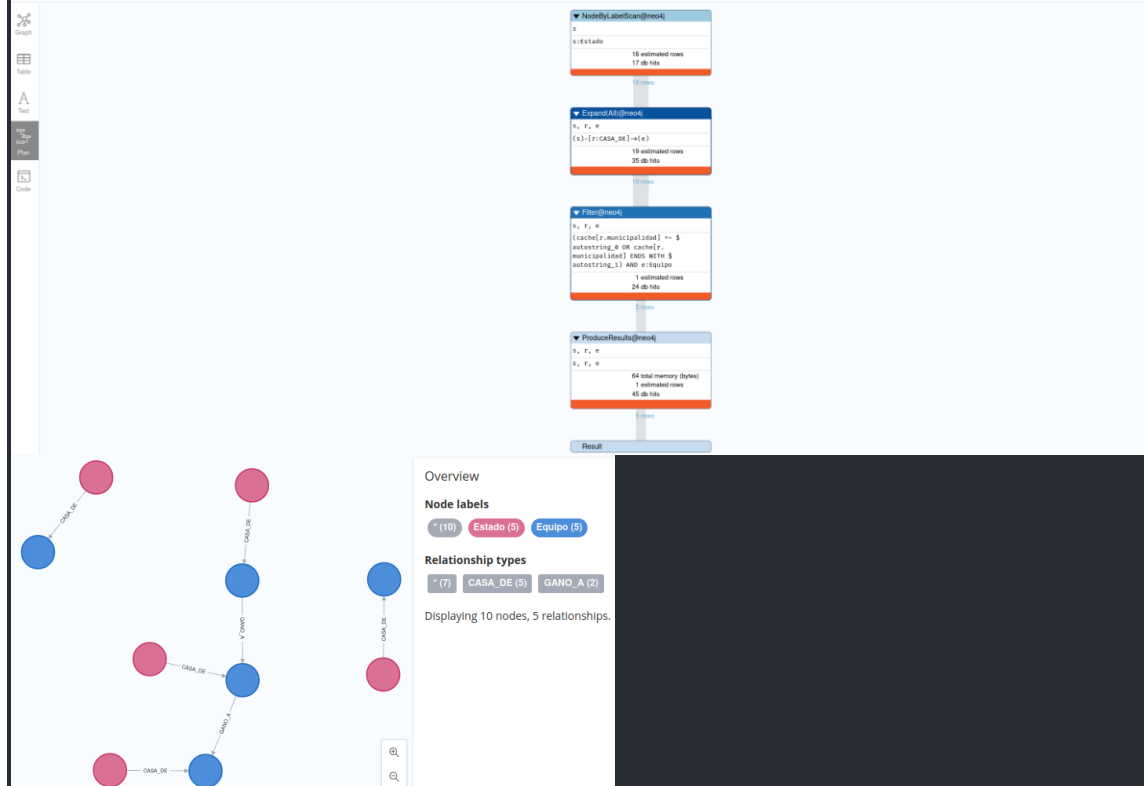
```
MATCH (s:Estado)-[r:CASA_DE]->(e:Equipo)
```

```
WHERE r.municipalidad =~ '[F|f].*'
```

```
OR r.municipalidad ENDS WITH 'y'
```

```
RETURN s, r, e; // No usa el índice
```

```
neo4j$ PROFILE MATCH (s:Estado)-[r:CASA_DE]->(e:Equipo) WHERE r.municipalidad =~ '[F|f].*' OR r.municipalidad ENDS WITH 'y' RETURN s, r, e;
```



```
PROFILE
```

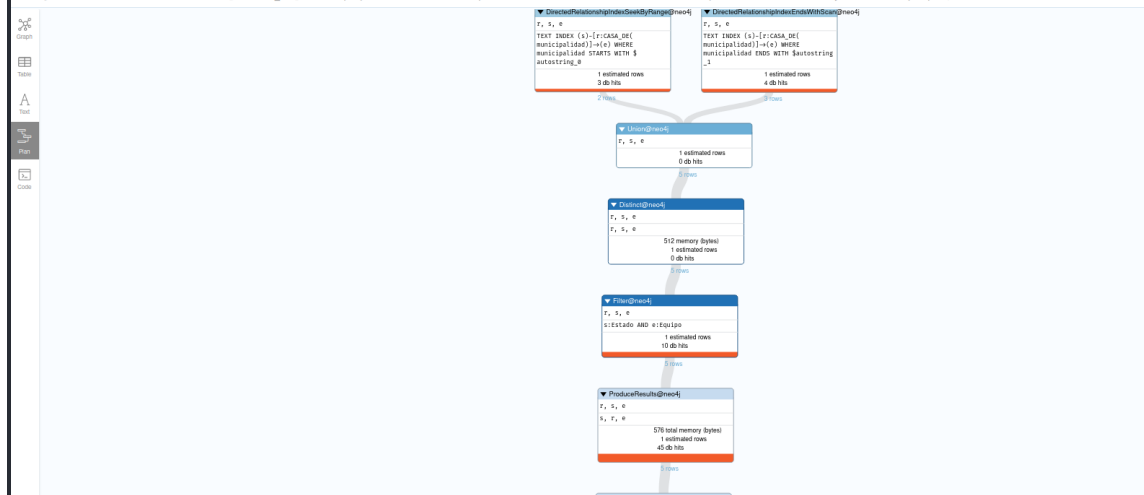
```
MATCH (s:Estado)-[r:CASA_DE]->(e:Equipo)
```

```
WHERE r.municipalidad STARTS WITH 'B'
```

```
OR r.municipalidad ENDS WITH 'y'
```

```
RETURN s, r, e; // Sí usa el índice
```

```
neo4j$ PROFILE MATCH (s:Estado)-[r:CASA_DE]->(e:Equipo) WHERE r.municipalidad STARTS WITH 'B' OR r.municipalidad ENDS WITH 'y' RETURN s, r, e;
```



The screenshot displays the Neo4j web interface. On the left, a graph visualization shows nodes (blue and pink circles) connected by relationships (labeled 'CASA_DE' and 'GANQ_A'). On the right, an 'Overview' panel shows statistics: 10 nodes, 5 relationships, and 5 relationship types. Below the graph, a Cypher query is executed in the 'neo4j\$' shell:

```
// player_name_idx
CALL db.index.fulltext.queryNodes("player_name_idx", "kelce") YIELD node, score
RETURN node.nombre, score;
```

The result is shown in a table with two columns: 'node.nombre' and 'score'.

	node.nombre	score
1	"Travis Kelce"	1.0711548328399658
2	"Jason Kelce"	1.0711548328399658

Conclusiones

El uso de los índices potencializaría en grandes proporciones el desempeño de las consultas en todo tipo de aplicaciones, especialmente en aquellas que cuenten con búsquedas de texto. A pesar de que son muchos los beneficios que se pueden encontrar con este elemento de Neo4j, su abuso también puede resultar dañino, dado que el espacio en memoria que implican es algo que debe ser administrado con suma cautela para evitar métricas indeseadas en la operación real de las bases.

Los índices de rango no operan de buena forma en textos demasiado complejos, es decir, están limitados a cierta funcionalidad, por ello, son preferibles en valores numéricos, como ha sido el caso del que se implementó en el desarrollo de esta actividad.

Para la parte de los índices de texto, la gran diferencia entre un índice de texto normal y un índice “full-text” es la manera en que se analizan las cadenas de caracteres. Con el primero, las operaciones de búsqueda se limitan a búsquedas muy específicas, las cuales también son muy útiles, pero no ofrecen demasiada flexibilidad; por otro lado, con el soporte de “Apache Lucene”, los campos de texto se descomponen en cadenas más pequeñas o según como se le indique al motor de consultas.

Las bondades de cada tipo de índice solo pueden ser vistas cuando se aplican correctamente y se usan para lo que los desarrolladores indican que son útiles. No se debe esperar un gran desempeño de índices comunes sobre campos con cadenas de texto inmensas, sin embargo, incluso habría que reconsiderar en primera instancia si una base de grafos es lo ideal para almacenar amplios volúmenes de caracteres...