Base de datos basado en grafos de investigadores de Concytec para un análisis de diversas perspectivas

Por
Rodrigo Alexander Mamani Sucacahua

Tesis presentada a la
Escuela Profesional de Ciencia de la Computación
de la
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
para el curso de Proyecto Final de Carrera I

Capítulo 1

Introducción General

1.1. Introducción

Las universidades públicas y privadas presentan la necesidad de contactar,contratar y encontrar investigadores en áreas en específico, ya sea para ayudar a los alumnos en su asesoramiento para sus proyectos de tesis o para dirigir proyectos de pregrado, para estos fines se evaluan de diversas maneras y puntos de referencia dados por los interesados. Esta toma de decisiones ayuda a los interesados tener veracidad y respaldo de calidad en el área de investigación.Para esto se opta a buscar en bases de datos públicas las cuales brindan dicha información realizando largas búsquedas para encontrar a los investigadores de su interés. Concretamente en el Perú la página de DINA de Concytec brinda información de los investigadores como bibliografía, experiencia profesional, datos académicos, producción científicas, proyectos de investigación entre otros dotas los cuales pueden ser de gran utilidad para dicho objetivo .

1.2. Definición del problema

Las búsquedas largas en páginas como DINA son muy engorrosas ya que no solo es encontrar a un sector de investigadores de su interés sino también leer los datos en forma de texto y sus curriculum vitae respectivos. Además, existen datos abstractos del interés académico que no se muestran en dicha página las cuales las relaciones de estas a menudo son hasta necesario consultar con el mismo investigador para saber si tiene lo que se busca o no, como que si tiene contacto con el extrangero, si conoce a colegas o investigadores de su rama entre otras cosas.
Al plantear el traslado de los datos a una base de datos lo mas común es hacerlo a una base de

datos relacional en lenguaje SQL , sin embargo, al momento de hacer consultas para encontrar un dato determinado se usan demasiados comandos de consultas JOIN , los cuales, si bien es cierto que es mas fácil la interacción con datos existen bases de datos que pueden brindar los datos requeridos de manera óptima como una base de datos basada en grafos.

1.3. Objetivos

1.3.1. Objetivo General

- Analizar el funcionamiento de la base de datos basada en grafos de DINA.

1.3.2. Objetivos Específicos

- Realizar el traslado de datos de la página de DINA a una base de datos basada en gráfos
- Usar Neo4j para probar el funcionamiento con datos de investigadores de la UNSA
- Realizar un análisis de algunas relaciones hechas en cypher (lenguaje de consultas de Neo4j)
- Mostrar gráfos de las consultas obtenidas.

1.4. Hipótesis

La herramienta Neo4j es muy útil para la creación y el análisis base de datos.

1.5. Estado del Arte

El enfoque dado para el muestreo de investigadores es eficiente para la búsqueda concreta de los investigadores de interés y se demuestra el funcionamiento de esta base de datos basa den grafos y cuan eficiente es este modelo [Afandi and Wahyuni, 2020]. Dina no es la única fuente de datos de investigadores a nivel mundial, en este artículo plantea la visualización de datos de DSPACE en Neo4j y Dephi herramientas de visualización de código de acceso libre y apliaciones desktop [Aryani et al., 2017]. Para tener un buen análisis de una herramienta de creación de base de datos basado en grafos como Neo4j se debe evaluar como el back-end trabajado en esta aplicaciónayuda a los desarrolladores para su fácil entendimiento[Holzschuher and Peinl, 2013].

Para analizar a fondo la creación de la base de datos se deben evaluar la relación entre atributos y la asociación que estos presentan con la base de datos.[Lu et al., 2017].También se deben evaluar las consultas y transacciones que se pueden realizar en la base de datos creada en Neo4j[Vukotic et al., 2015].
Neo4j al ser una herramienta basada en grafos, es evidente que la base de datos creada sea basada en grafos, esta puede ser planteada en una base de datos distribuida como las social networks la cual ,tal como la base de datos general, de Neo4j esta también puede ser visualizada de manera óptima[Agudo Merino, 2020].
Como toda aplicación Neo4j tiene ventajas y desventajas en su uso, las cuales mediante comparaciones con diversas aplicaciones pueden ayudar al usuario a decidir si es factible o no usar dicha herramienta[Fernandes and Bernardino, 2018].Las limitaciones que esta herramienta presenta pueden ser evaluados por su interacción de la base de datos con las consultas y con los datos los cuales demuestran que Neo4j es una herramienta optimizable[Miller, 2013].
El análisis de una herramienta como Neo4j también puede ser evaluado por el tiempo de demora de la creación de la base de datos y su complejidad de consultas comparadas con otra herramienta como PostgreSQL.[Stothers and Nguyen, 2020]. Neo4j no solo sirve para bases de datos convensionales , sino también ayudan para el mejor control de la semántica e interacción de las bases de datos como las redes sociales y web semántica[Guia et al., 2017].

1.6. Marco Teórico

1.7. Variables de investigación

1.7.1. Variables Independientes

- Análisis
- Perspectivas

1.7.2. Variables Dependientes

- Base de datos basado en grafos
- Investigadores de Concytec

Capítulo 2

Técnicas a reproducir

2.1. Base de datos basado en Neo4j

2.1.1. Abstract

Con el desarrollo de Internet, el crecimiento de los datos cinematográficos rápidamente, la relación entre los datos de la película se vuelve más y más complicado. Las relaciones entre películas, actores y los escritores son información importante tanto para los productores de cine y audiencias. El sitio web de datos de películas no solo necesita almacenar vídeos de películas, también necesitan almacenar información sobre directores, escritores, actores, etc. Si tales datos se almacenan en una base de datos relacional, el las conexiones entre diferentes tablas se pueden establecer a través de llave. Sin embargo, cuando hay muchas relaciones, la base de datos relacional tradicional no solo tiene una gran cantidad de redundancia de datos, pero también se vuelve difícil de actualizar dinamicamente. Además, es difícil realizar consultas complejas relaciones entre dos entidades. Por ejemplo, cuando quieren saber si los principales productores de las dos películas tienes una coproducción de otras películas, necesitas encontrar otras películas producidas por los principales productores de las dos películas, y luego analice si existe una intersección entre ellos. Por lo tanto, la base de datos no relacional es una buena elección para investigar y procesar datos de películas. Neo4j, que es un excelente herramienta de base de datos de gráficos, almacena datos en forma de gráfico que puede representar objetos con nodos, aristas y propiedades [Lu et al., 2017]. En consecuencia, es adecuado para almacenar complejos y dinámicos relaciones entre objetos de datos fílmicos, por lo que se desarrollará la recreación de la base de datos no relacional basada en grafos de Neo4j de DataFilm y su respectivo análisis.

2.1.2. Neo4j

Al momento de usar o crear bases de datos, las bases de datos relacionales han sido dominantes durante mucho tiempo. Con la aplicación de Wed 2.0, el auge de las redes sociales, la dependencia y la complejidad de los datos internos aumentan gradualmente, Cada vez surgen más problemas en las bases de datos relacionales. Luego, apareció la base de datos basado en grafos.
En los últimos años ha habido una enorme cantidad de bases de datos basado en grafos de alto rendimient, existen multiples herramientas que trabajan con el diseño y la arquitectura de dicha base de datos entre ellas tenemos a Neo4j. Esta herramienta en específico trabaja en la corriente principal de un software de código abierto basado en Jav, actualmente su kernel es un motor de gráficos muy rápido, con la recuperación de datos, con dos fases del envío, soporte para transacciones XA y otras características del producto de base de datos. Neo4j es una base de datos orientada a la red, es decir, una base de datos integrada con motor de persistencia Java totalmente transaccional basado en disco que almacena datos estructurados en redes en lugar de tablas. El lenguaje de esta herramienta es llamado Cypher el cual presenta sus propios comandos y formas de crear una base de datos con sus respectivos atributos. En este modelo, los datos de dominio se expresan en un .espacio de nodos"la cual esta compuesta de: una red de nodos, relaciones y propiedades (pares clave-valor), en comparación con las tablas del modelo relacional, que trabajan con filas y columnas. Las relaciones son objetos de primera clase y pueden también se tener propiedades, revelando el contexto de cuales son los nodos con los que se interactúan.

2.1.3. Creación de la base de datos

Para crear una base de datos basada en grafos debemos saber el funcionamiento de este y como empezar a crearla.La asociación es similar al borde en un grafo dirigido,El borde de esta base de datos consta de tres elementos: el nodo inicial, el nodo final y el tipo. La orientación del borde aclara aún más la semántica y la relación entre nodos.En la Figura 2.1 vemos que los atributos de los nodos y las relaciones se pueden definir por clave-valor. Actores, directores, escritores y películas son entidades diferentes cuando los datos de la película son almacenados, la base de datos de Neo4j no solo necesita almacenar entidades, sino también necesita almacenar las relaciones entre entidades.

```
Create
(TomHanks:Stars{name:'Tom Hanks',born:1956}),
(ForrestGump:Movie{title:'Forrest Gump', released:1994}),
(TomHanks)-[:ACTED_IN]->(ForrestGump)
```

Figura 2.1: Ejemplo de la creación de un borde [Lu et al., 2017].

2.1.4. Consultas y resultados

En la Figura 2.2 vemos el grafo de la película <The Green Mill>como el punto de partida, se crea una pequeña base de datos con 19 nodos y 19 relaciones como se muestra en la siguiente figura. En la figura tambien veremos que, el nodo de color verde significa que es un nodo de película,el amarillo significa que la etiqueta pertenece al nodo del actor,la etiqueta rosa significa que es un escritor, etiqueta azul significa que es un director, se utilizan diferentes colores para representar diferentes etiquetas y por ende diferentes tipos de nodos a construir.

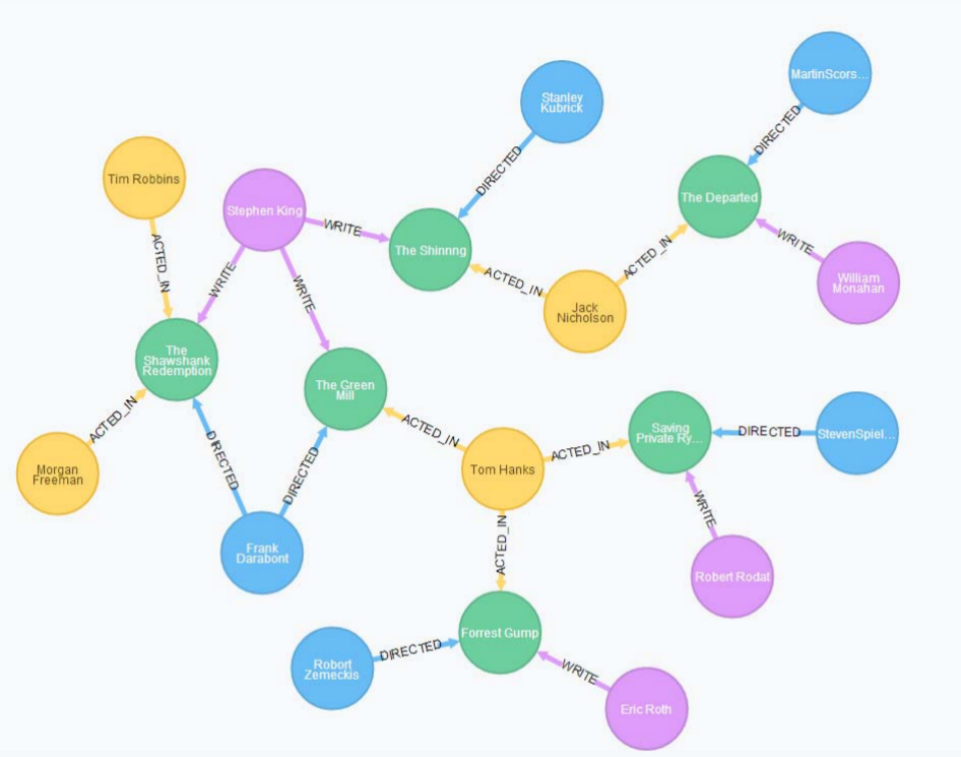

Figura 2.2: Base de datos de películas en formato grafo [Lu et al., 2017].

Aquí hay un ejemplo,en la Figura 2.3 según el usuario ha visto la película. <La redención Shawshank>y <Forrest Gump>, el sistema recomienda otras películas que se relacionen con las dos películas al usuario. Por análisis encontramos que Tom Hanks actuó en <Forrest Gump>y <The Green Mill>, el director Frank Darabont no solo dirigió <THE Green Mill>pero también dirigió <The Shawshank Redención>, por lo que encontramos la relación potencial entre la película <Forrest Gump>, <The Shawshank Redemption>y <The Green Mill>, por lo que podemos recomendar la película <The Green Mill>al usuario. La siguiente es la consulta en el lenguaje Cypher muestra la consulta para este ejemplo.

```
match movie1
where movie1.title="The Shawshank Redemption"
match movie2
where movie2.title="Forrest Gump"
match
searchPath=movie1<-[*]-(people1)-[*]->movie-[*]-(people
2)-[*]->movie2
return  distinct movie.title,searchPath
```

```
return 1 line, times 20 ms
```

Figura 2.3: Consulta en lenguaje Cypher [Lu et al., 2017].

Ya que la herramienta Neo4j nos brinda la opción de representar nuestra base de datos en una arquitectura de grafos usaremos esta opción para una mejor interpretación y después veremos el gráfico de dicha consulta en la Figura 2.4 y como esta representado sus relaciones y nodos correspondientes.

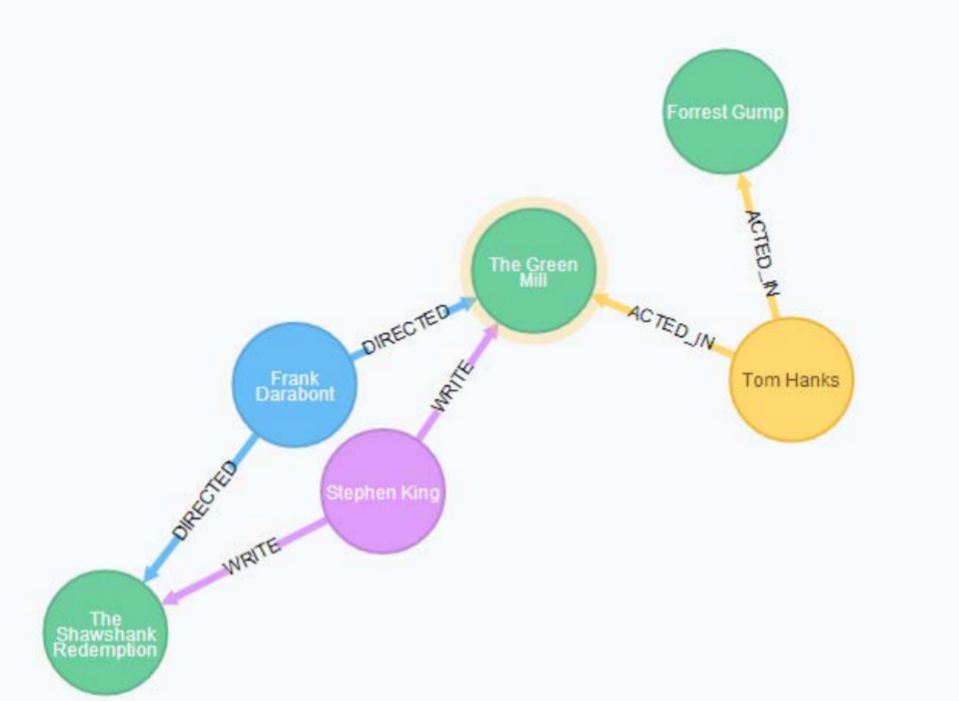

Figura 2.4: Representación en grafos de la consulta anterior [Lu et al., 2017].

2.2. Base de datos de investigación universitaria en Neo4j

2.2.1. Abstract

En su mayoría, los datos relacionados con la investigación se modelan usando una base de datos relacional optimizada para el proceso de transacciones, esto se debe a que es el tipo de base de datos mas común para dicho trabajo. En muchos casos, esta solución es eficaz y eficiente, lo suficiente como para responder consultas básicas e transacciones sencillas. Sin embargo, cuando los usuarios solicitan un análisis más detallado, más expansivo, con múltiples perspectivas y, a veces, un mayor análisis abstracto, la base de datos relacional lucha por proporcionar respuestas. Este estudio propone una base de datos basado en grafos de investigación implementado usando neo4j como la herramienta de creación para responder a los problemas.La base de datos permite que la universidad pueda analizar el trabajo individual y colaborativo de los investigadores en conjunto de compañeros investigadores dentro y fuera de las universidades. El estudio concluye que el gráfico de investigación que muestra que la implementación de la base de datos es más

eficiente para responder preguntas que la implementación de la base de datos relacional.

2.2.2. Métodología

Para la recolección de datos para la creación de nuestra base de datos basada en grafos se optó por evaluar las multiples fuentes de información consistentes y de buena procedencia que se pueden encontrar en la nube. Entre aquellas podemos encontrar a Google Scholar.
Google Scholar es una plataforma denominado SINTA (Índice de Ciencia y Tecnología). Ristekbrin (Ministerio de Investigación y Tecnología / Investigación Nacional y Innovation Body) de Indonesia actualmente alberga SINTA. SINTA utiliza datos de Google Scholar y otros fuentes y luego calcula las capacidades de los investigadores, instituciones y revistas en Indonesia. Los resultados son entonces puesto a disposición del público de forma gratuita. Google Scholar presenta ventajas que vale la pena destacar com los hechos que Google Scholar ofrece referencias académicas más actualizadas. Existen también API (interfaces de programación de aplicaciones) en varios plataformas de programación creadas por desarrolladores que los investigadores, profesionales y el público en general pueden utilizar para conectarse a Google Scholar y recuperar datos de él de forma gratuita para académicos con fines de investigación.

2.2.3. Resultados

Primeramente mostramos la creación de nuestra base de datos basada en grafos y usando como fuente de información la plataforma de google scholar. La base de datos de grafos resultante de este estudio de caso consiste de 12 etiquetas de nodo, 15 relaciones y 33 claves de propiedad. La Tabla 2.5 muestra todos los nombres de las etiquetas de los nodos, y la Tabla 4.4 muestra los relación entre nodos.

No	Node Label	Description
1	Affiliation	It represents a researcher's affiliation, such as a university or a research institution, usually written in a scientific article.
2	Article	It represents scientific articles or reports written by Researchers
3	DbIndexer	It describes indexing database services of scientific articles provided by specific organizations, such as Scopus, Web Of Science.
4	IntelPropRight	It typifies any intellectual property right owned by any Researcher
5	Organization	It represents any organization of any kind which researchers affiliate with, receive projects grant from, and have their intellectual property approved
6	Person	It depicts any person in general who can be a researcher or any other type of role or job
7	Publisher	It portrays a particular type of organization which offers publication products
8	Pubname	It illustrates any publication product offered by a Publisher
9	QualityMeasure	It represents any quality measurement of a publication product offered by an indexing database service either directly offered by the database or indirectly by a third party
10	RegisterOffice	A particular type of organization which provides intellectual property rights services from applying to awarding
11	ResearchProject	Any project which can be either research or community engagement type, self-funded or granted by an individual or a group of organizations
12	Researcher	It represents a select type of Person who works either partially or entirely as a researcher.

Figura 2.5: Nodos de la base de datos de investigación [Afandi and Wahyuni, 2020].

No	Node	Relation (Action) Name	Node
1	Researcher	AFFILIATES_WITH	Affiliation
2	Researcher	AUTHORS	Article
3	Researcher	CO_AUTHORS	Article
4	Researcher	CO_LEAD	ResearchProject
5	Researcher	CO_OWNS	Intellectual PropertyRight
6	Organization	GRANTS	ResearchProject
7	RegisterOffice	GRANTS_IPR	Intellectual PropertyRight
8	DbIndexer	HAS_MEASUREMENT	QualityMeasure
9	Article	INDEXED_IN	DbIndexer
10	Researcher	LEAD	ResearchProject
11	Researcher	OWNS	Intelletual PropertyRight
12	ResearchProject	PRODUCES	Intellectual PropertyRight
	ResearchProject	PRODUCES	Article
13	Article	PUBLISHED_IN	PubName
14	PubName	RANKED	QualityMeasure
15	Researcher	REG_AUTH_IN	DbIndexer

Figura 2.6: Relaciones basado en los nodos de la base de datos de investigación [Afandi and Wahyuni, 2020].

La consulta de grafos escrita en Cypher como se muestra en la Figura 2.7 es eficiente y simple. Podría decirse que puede ser formulado y realizado por trabajadores del conocimiento con menos experiencia que recibieron la formación previa y equipado con información tal como nodo y relación de nodo. Además, la información visual que se muestra también es informativa. y adecuado para diferentes niveles de espectadores, desde el conocimiento trabajadores a ejecutivos. Una consulta más larga para responder a la misma pregunta en un relacional es simulada en la Figura 2.8. La diferencia es clara. En una base de datos relacional, el requisito debe ser muy explícito por adelantado para que se traduzca correcta y cuidadosamente en consultas antes de decidir qué tablas unir en qué columnas y qué campos deben recuperarse. De hecho, requiere algunos nivel de experiencia técnica para ejecutarlo.

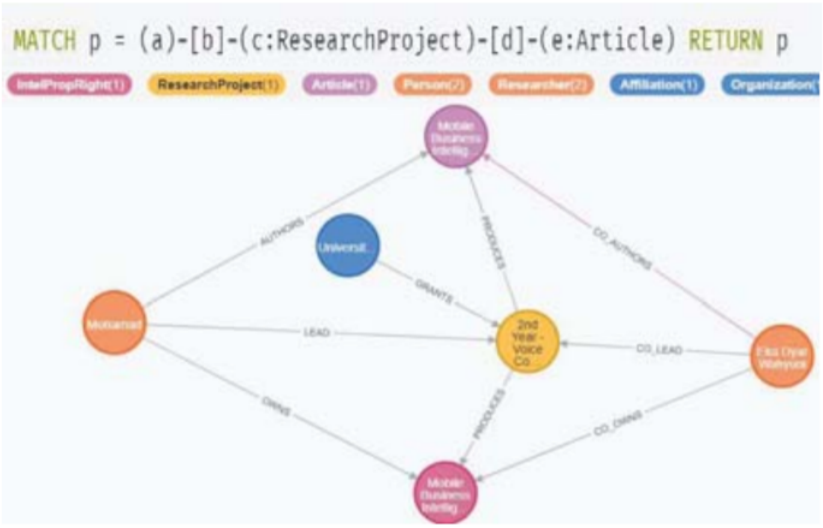

Figura 2.7: Consulta en lenguaje Cypher para la base de datos basada en grafos [Afandi and Wahyuni, 2020].

```
SELECT  a.name,
        rp.title,
        p.name,
        rpm.role,
        ar.title,
        aa.role
FROM  affiliation a
JOIN research_project rp on (rp.id_affiliation = a.id)
JOIN research_project_member rpm (rp.id_project =
rpm.id_project)
JOIN article ar on (ar.id_project = rp.id_project)
JOIN article_authors aa on (ar.id_art = aa.id_art)
JOIN person p on (p.id_person = aa.id_person) and
(p.id_person = rpm.id_person)
```

Figura 2.8: Consulta en SQL para la base de datos relacional [Afandi and Wahyuni, 2020].

Capítulo 3

Marco teórico

3.1. Antecedentes

3.1.1. Bases de datos

Un aspecto clave del diseño de una base de datos es asegurar que el La base de datos contiene solo valores de datos válidos consistentes con semántica de la base de datos incluso en presencia de:

- Inserciones
- Eliminaciones
- Actualizaciones

Las restricciones simplifican las aplicaciones del sistema de base de datos realizar comprobaciones para garantizar la validez de los datos y consistencia.
Esta verificación centralizada también es más confiable ya que la aplicación puede no realizar una o más de las comprobaciones.

Una base de datos solo debe contener datos válidos y consistentes en todo momento con la posible excepción en medio de actualizaciones [Gehani, 1990]. Algunos tipos de validez de datos se pueden garantizar mediante SQL limitaciones. Ejemplos:

- Los estados se identifican utilizando sus abreviaturas de dos letras.
- Los valores deben estar dentro de un rango específico.
- Los campos de datos son obligatorios y no se pueden omitir.

- Los valores de una columna deben ser únicos.

Ejemplos de restricciones complejas de bases de datos:

- Las abreviaturas de estado de 2 letras representan abreviaturas de estado válidas.
- Los códigos postales representan códigos postales válidos y coinciden con las direcciones especificadas.

3.1.2. Introducción a Base de datos basada en grafos

Un gráfico es una estructura de datos compuesta por aristas y vértices. La tecnología de base de datos de gráficos es una herramienta eficaz para modelar datos cuando un enfoque en la relación entre entidades es una fuerza impulsora en el diseño de un modelo de datos.El Modelado de objetos y las relaciones entre ellos significan que casi cualquier cosa se puede representar en un gráfico correspondienteen un gráfico común. El tipo admitido por la mayoría de los sistemas es el gráfico de propiedades. Los gráficos de propiedades son gráficos múltiples atribuidos, etiquetados y dirigidos [Miller, 2013].

Ejemplo de BD basada en grafos

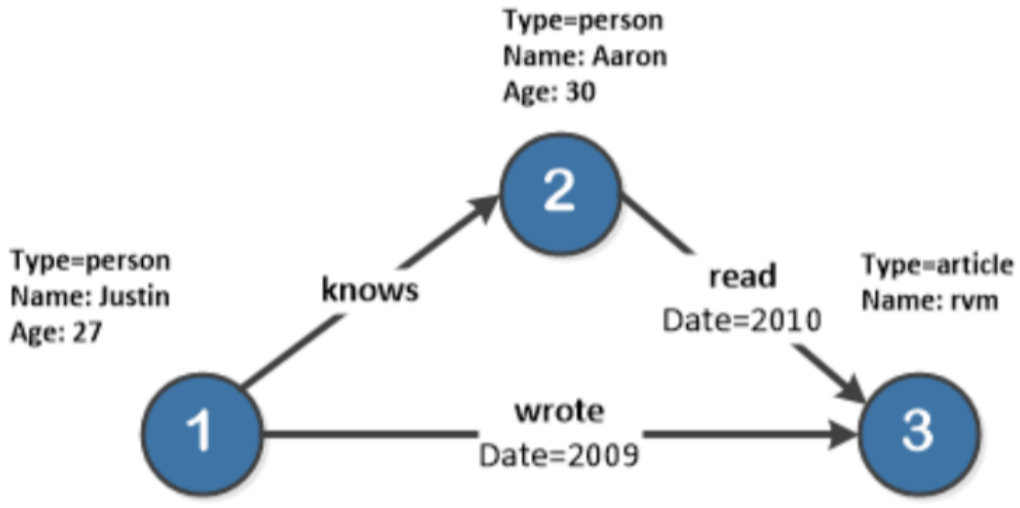

Figura 3.1: Ejemplo de gráfico de BDBG Fuente: [Miller, 2013].

A pesar de las limitaciones involucradas en el uso de un esquema, existen muchos beneficios, algunos de los cuales superan las restricciones implícitas en el RDBMS. Estas bases de datos están bien respaldadas por sus respectivos proveedores, están probadas en batalla y sus las fortalezas / limitaciones son bien conocidas, tienen un lenguaje de consulta común (SQL), hay una amplia reserva de talentos de capacitados profesionales a los que recurrir, y estas plataformas están muy bien documentadas. Por el contrario, las bases de datos de gráficos se enfrentan a una conjunto de desafíos inherentes a su diseño.

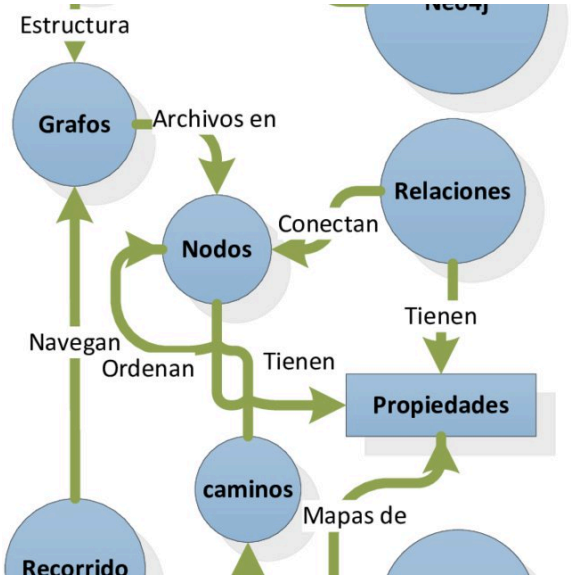

Figura 3.2: Ejemplo de arquitectura de una BD basada en grafos.

3.1.3. Aplicación de base de datos basado en grafos

Las bases de datos gráficas realmente brillan cuando se trabaja en áreas donde la información sobre la interconectividad o la topología de datos es importante. En tales aplicaciones, las relaciones entre los datos y los datos en sí suelen estar al mismo nivel . Muchas empresas tienen desarrolló implementaciones internas para hacer frente a la necesidad de sistemas de bases de datos de gráficos[Miller, 2013].
Ejemplos serían:

- Open Graph de Facebook.
- Knowledge Graph de Google.
- FlockDB de Twitter.

3.1.4. Introducción a Neo4j

Aunque los gráficos se utilizan ampliamente durante el proceso de desarrollo de software, los desarrolladores tienden a olvidarse de los gráficos cuando se trata de la persistencia de datos. Intentamos ajustar los datos en tablas y columnas relacionales, y para normalizar y renormalizar su estructura hasta que se vea completamente diferente de lo que está tratando de representar. Una lista de control de acceso es un ejemplo. Este es un problema resuelto una y otra vez nuevamente en muchas aplicaciones empresariales. Por lo general, tendría tablas para usuarios, roles y recursos. Luego, tendría tablas de muchos a muchos para asignar usuarios a roles y roles a los recursos. Al final, tendrías al menos cinco tablas relacionales para representar un estructura de datos simple, que en realidad es un gráfico. Entonces usarías un objeto relacional herramienta de mapeo (ORM) para mapear estos datos a su modelo de objeto, que también es un gráfico.
¿No sería bueno si pudiera representar los datos en su forma natural, haciendo mapeos más intuitivos y omitiendo el proceso repetido de "traducir"los datos hacia y desde un motor de almacenamiento?.
Gracias a las bases de datos de gráficos, puede hacerlo. Bases de datos de gráficos utilizar el modelo de gráfico para almacenar datos como un gráfico, con una estructura que consta de vértices y bordes, las dos entidades utilizadas para modelar cualquier gráfico. Además, puede utilizar todos los algoritmos de la larga historia de la teoría de grafos para resolver problemas de gráficos de manera más eficiente y en menos tiempo que usando consultas de bases de datos relacionales. [Vukotic et al., 2015].

3.1.5. Graficar datos en Neo4j

Neo4j almacena datos como vértices y bordes o, en terminología de Neo4j, nodos y relaciones. Los usuarios se representarán como nodos y las amistades se representarán como relaciones entre los nodos de usuarios. Si echa otro vistazo a la red social en la Figura 3.3, verá que no representa más que un gráfico, con los usuarios como nodos y las friendship como relaciones. Existe una diferencia clave entre las bases de datos relacionales y Neo4j, que se encuentran de inmediato: consulta de datos. No hay tablas ni columnas en Neo4j, tampoco hay comandos de selección y combinación basados en SQL. Entonces, ¿cómo se consulta un base de datos de gráficos? Todos los gráficos de bases de datos, toma un poderoso concepto matemático que denota que la respuesta no es escribir una función MapReduce distribuida. Neo4j, lo muestra como teoría de grafos y lo usa como un motor poderoso y eficiente para consultar datos. Este concepto es el gráfico transversal y es una de las herramientas principales que hace que Neo4j sea tan poderoso para tratar con gráfico

de datos [Vukotic et al., 2015].

Depth	Execution time (seconds) for 1,000 users	Count result
2	0.028	~900
3	0.213	~999
4	10.273	~999
5	92.613	~999

Figura 3.3: Tiempos de ejecución para múltiples consultas de combinación usando un motor de base de datos MySQL en un conjunto de datos de 1.000 usuarios

3.1.6. Neo4j en el espacio NoSQL

Desde el comienzo del software de computadora y los datos que las aplicaciones han tenido que tratar con ha crecido enormemente en complejidad. La complejidad de los datos incluye no sólo su tamaño, también su interconexión, su estructura en constante cambio y el acceso concurrente a los datos. Con todos estos aspectos del cambio de datos, se ha reconocido que las bases de datos relacionales, que durante mucho tiempo han sido el estándar de facto para el almacenamiento de datos, no son las más adecuadas para todos los problemas que requieren los datos cada vez más complejos. Como resultado, se ha creado una serie de nuevas tecnologías de almacenamiento, con el objetivo común de resolver los problemas en los que las bases de datos relacionales no son buenas. Todas estas nuevas tecnologías de almacenamiento caen bajo el término general NoSQL.
Aunque el nombre NoSQL se atascó, no refleja con precisión la naturaleza del movimiento, dando la impresión (errónea) de que va en contra de SQL como concepto. Un mejor nombre probablemente sería el de bases de datos no relacionales, ya que el relacional y paradigma no relacional fue el tema de discusión, mientras que SQL es sólo un lenguaje utilizado con tecnologías relacionales. El movimiento NoSQL nació como un reconocimiento de que las nuevas tecnologías necesitaban para hacer frente a los cambios de datos. Neo4j, y las bases de datos de grafos en general, son parte del movimiento NoSQL, junto con una gran cantidad de almacenamiento más o menos relacionado con las tecnologías. Con los rápidos desarrollos en el espacio NoSQL, su creciente popularidad y la enorme cantidad de diferentes soluciones y tecnologías para elegir, cualquiera que sea nuevo en el mundo NoSQL enfrenta muchas opciones al seleccionar la tecnología adecuada [Vukotic et al., 2015].

3.1.7. Neo4j: la base de datos compatible con ACID

La gestión de transacciones ha sido un tema de conversación destacado en las discusiones sobre las tecnologías NoSQL desde que comenzaron a ganar popularidad. Negociar atributos transaccionales para un mayor rendimiento y escalabilidad ha sido un enfoque común en tecnologías no relacionales que apuntan a big data. Algunos (como BigTable, Cassandra y CouchDB) optó por compensar la coherencia, permitiendo a los clientes leer datos obsoletos en algunos casos en un sistema distribuido (consistencia eventual). En las tiendas de valores-clave que se concentraban en el rendimiento de lectura (como Memcached), la durabilidad de los datos no generaba mucho interés. Del mismo modo, la atomicidad a nivel de una sola operación, sin la posibilidad de envolver múltiples operaciones de bases de datos dentro de una sola transacción, es típico de las bases de datos orientadas a documentos. Si bien cada uno de los enfoques mencionados aquí son válidos en casos de uso específicos (como el almacenamiento en caché, altos volúmenes de lectura de datos, alta carga y simultaneidad), la falta de manejo de transacciones basado en ACID suele ser el primer obstáculo cuando se trata de introducir bases de datos no relacionales a cualquier empresa o entorno corporativo. Aunque ellos fueron ideados hace mucho tiempo para bases de datos relacionales, los atributos de transacción aún se reproducen una parte importante y fundamental en muchos casos prácticos de uso. Por tanto, Neo4j ha adoptado un enfoque diferente [Vukotic et al., 2015].

Capítulo 4

Experimentación

4.1. Técnica 1:Base de datos basado en Neo4j

4.1.1. Creación de la Base de datos

Creamos los scripts de inicialización de la base de datos de películas. Para esto en la Figura 4.1 vemos las característicasa de nuestra base de datos, desde los tipos de nodos y las relaciones hasta las propiedades que tienen cada nodo.

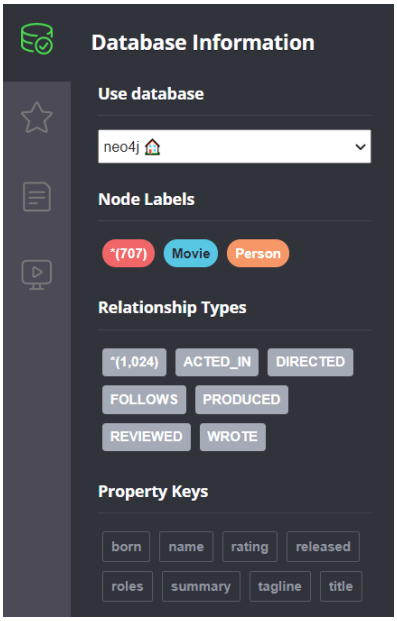

Figura 4.1: Datos generales de la base de datos de películas

A continuación en la Figura 4.2 muestra el grafo del pequeño segmento tomado para la muestra de nodo y relaciones correspondientes.

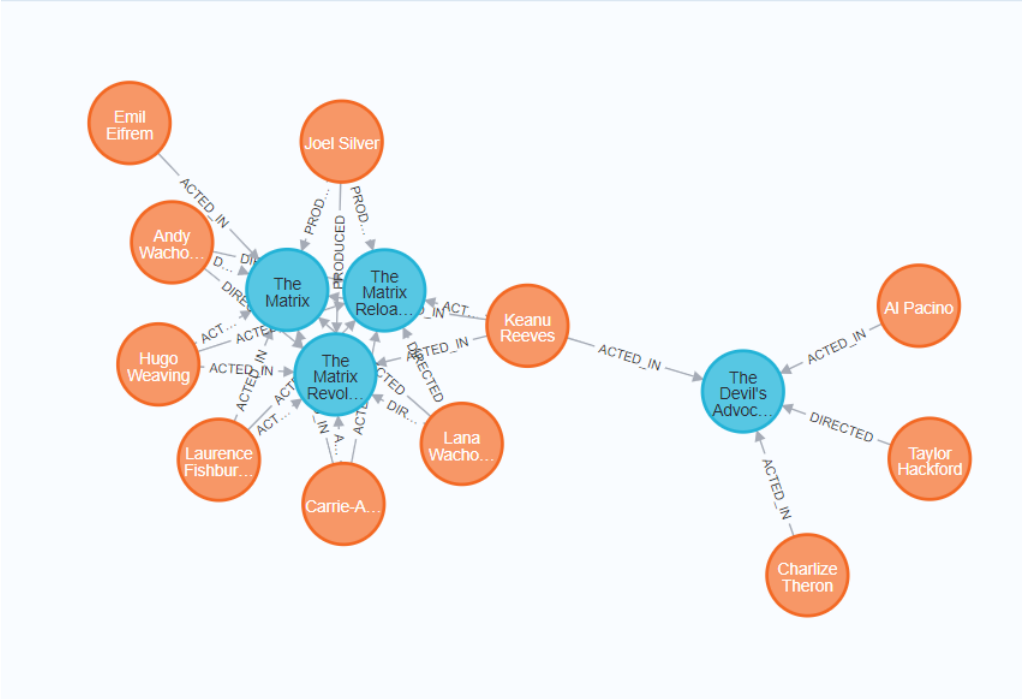

Figura 4.2: Gráfo de retorno según el tipo de nodo n como un pequeño segmento de toda la base de datos de películas

Para nuestro segundo tipo de nodo tenemos como tag a película en el cual la Figura 4.4 muestra un pequeño fragmento en forma de tabla de los nodos película y sus respectivos atributos:

- Tagline: Frase representativa de dicha película.
- Title: Nombre de la película y el año en el que fue lanzada.

Después, es mostrado en forma de nodos de grafo las películas que ingresamos a nuestra base de datos en la Figura 4.3

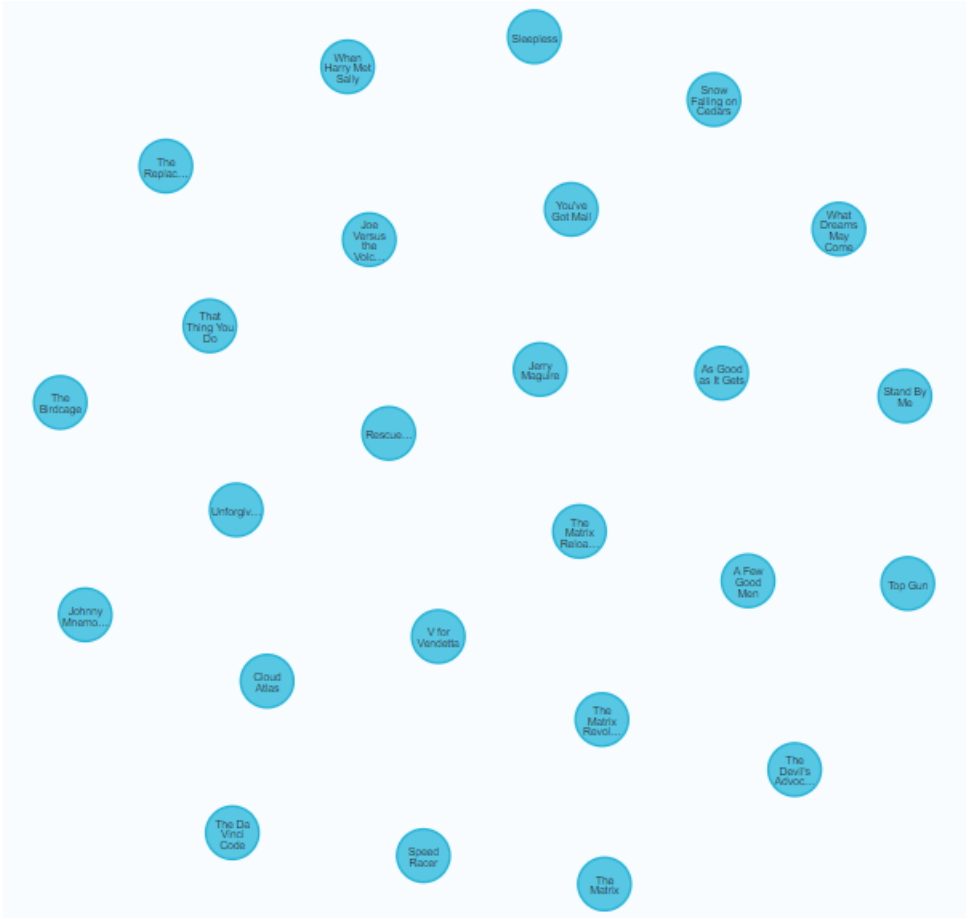

Figura 4.3: Grafo de nodos de películas sin relaciones ni conexiones.

```
{"tagline":"In every life there comes a time when that thing you dream
 becomes that thing you do","title":"That Thing You Do","released":199
6}

{"tagline":"Pain heals, Chicks dig scars... Glory lasts forever","titl
e":"The Replacements","released":2000}

{"tagline":"Based on the extraordinary true story of one man's fight f
or freedom","title":"RescueDawn","released":2006}

{"tagline":"Come as you are","title":"The Birdcage","released":1996}

{"tagline":"It's a hell of a thing, killing a man","title":"Unforgiven
","released":1992}

{"tagline":"The hottest data on earth. In the coolest head in town","t
itle":"Johnny Mnemonic","released":1995}

{"tagline":"Everything is connected","title":"Cloud Atlas","released":
2012}

{"tagline":"Break The Codes","title":"The Da Vinci Code","released":20
06}
```


Figura 4.4: Tabla de los nodos película de la base de datos

Finalmente mostramos el tercer tipo de nodo de nuestra base de datos de películas el cual tiene de etiqueta person , el cual brinda los datos y atributos como muestra en la Figura 4.5 que representan:

- Actores.
- Escritores.
- Directores.
- Productores.

Y de manera general la Figura 4.5 muestra como nodos sin conexiones ni relaciones los nodos person que existen en nuestra base de datos

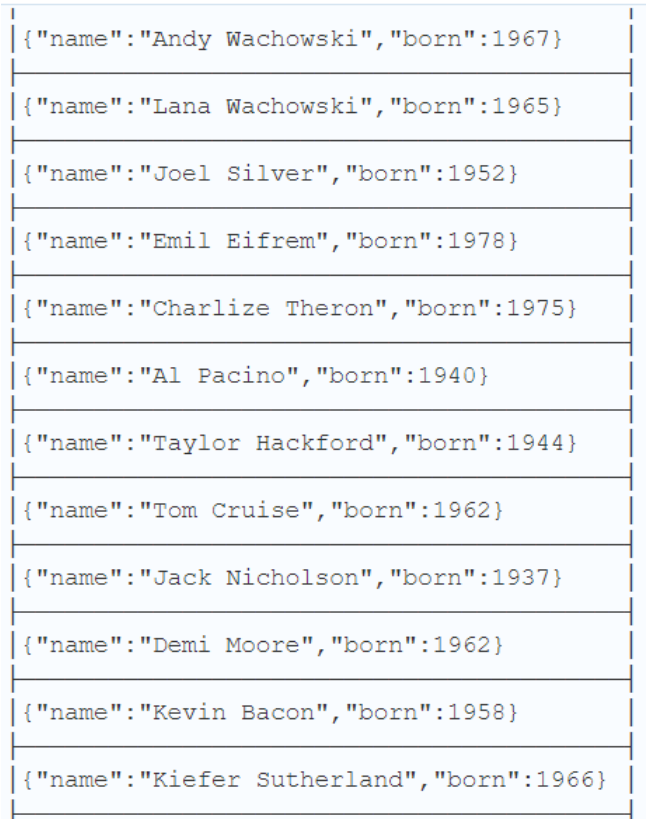

Figura 4.5: Tabla de los nodos person de la base de datos

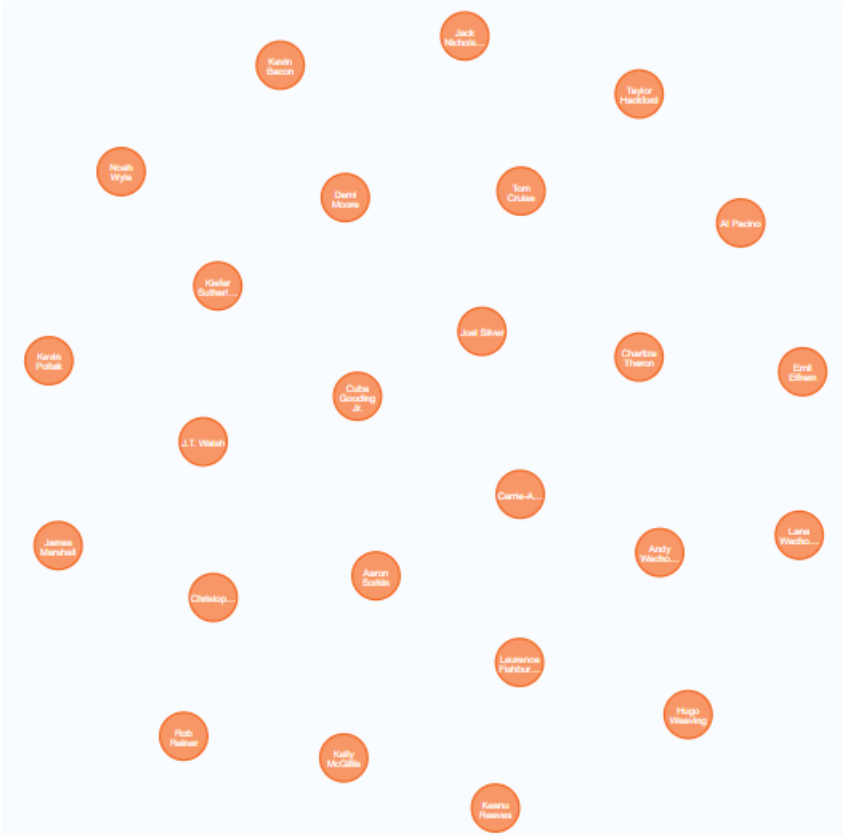

Figura 4.6: Nodos grafo con la etiqueta person de la base de datos de películas

4.1.2. Análisis de la base de datos

4.1.2.1. Consulta de los atributos de un nodo

El objetivo es mostrar como se realizan para mostrar un nodo en concreto usando los comandos de consulta del lenguaje Cypher.

En la Figura 4.7 vemos en la sintaxis de escritura que se usa MATCH para referirse a un conjunto de nodos de un tipo representado por una variable el cual con LIMIT definimos la cantidad de nodos a mostrar.

En la Figura 4.8 notamos que ya explicitamente escogemos el nodo de una película buscandola por el título de esta para mostrar el título y el año en el que fue realizado dicha película.

- Mostrar todos los nodos de un tipo en concreto , ya sea persona o película.

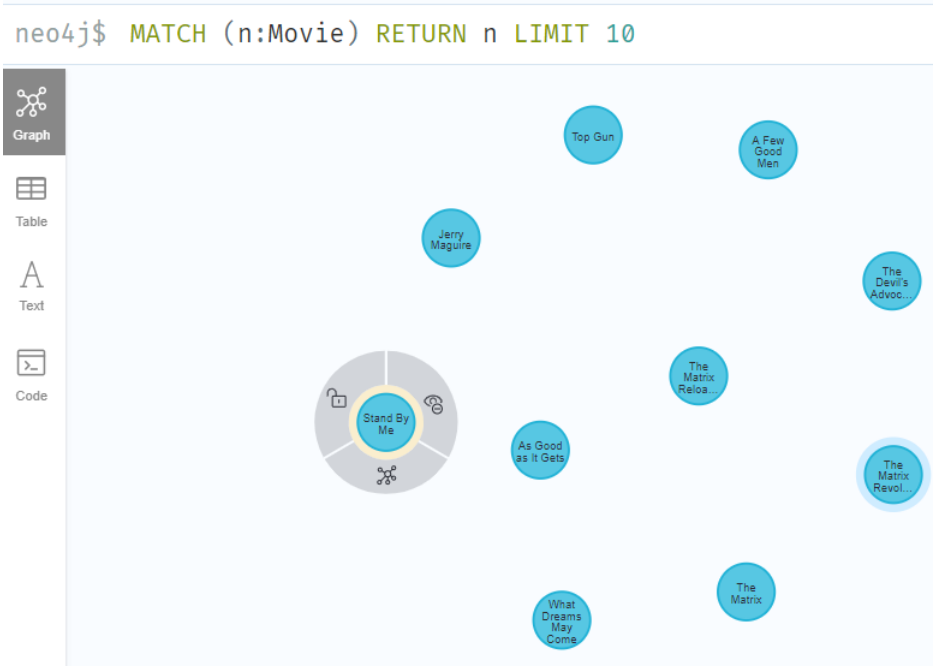

Figura 4.7: Nodos película mostrados en grafo limitado a 10

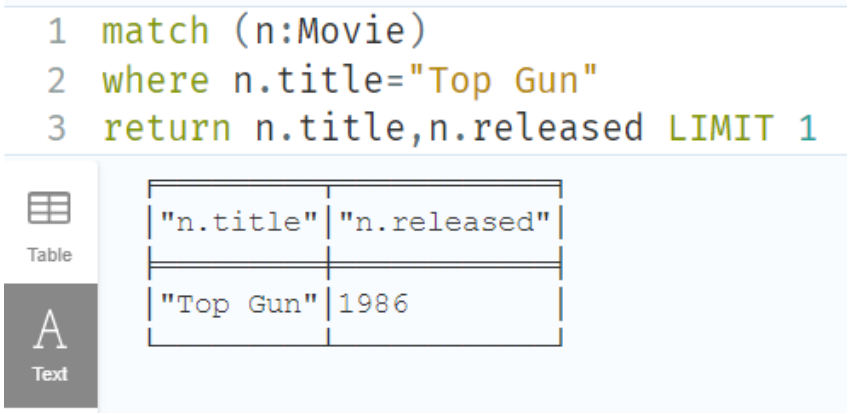

Figura 4.8: Nodo película con titulo de Top Gun y su año en el que fue realizado

- Mostrar un nodo y sus atributos en concreto En la Figura 5.4 vemos como un nodo película muestra sus relaciones con otros nodos en forma gráfica.

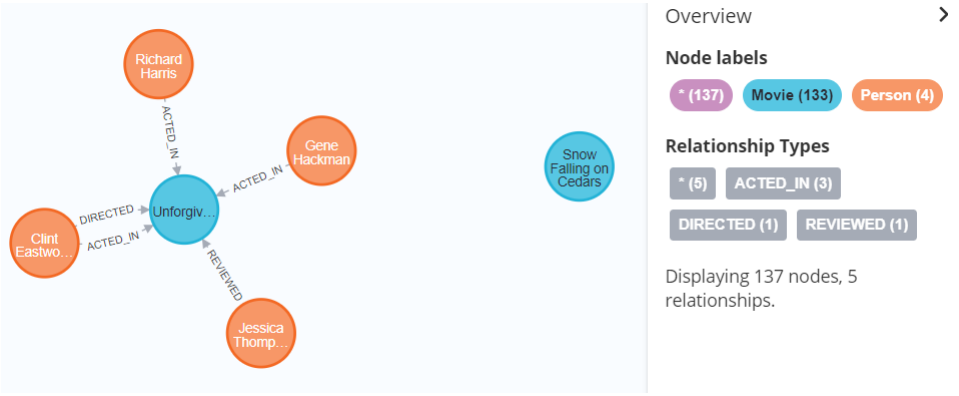

Figura 4.9: Nodo película Unforgiven con sus respectivas relaciones

4.1.2.2. Consulta encapsulada por diversos atributos

En la Figura 4.10 hacemos las búsquedas encapsuladas basadas en limitar el rango de búsqueda mediante un atributo en concreto.

```
MATCH (a:Movie) WHERE a.released > 1990 AND a.released < 1995 RETURN a LIMIT 10
```


{"tagline":"In the heart of the nation's capital, in a courthouse of the U.S. government, one man will stop at nothing to keep his honor, and one will stop at nothing to find the truth.","title":"A Few Good Men","released":1992}

{"tagline":"What if someone you never met, someone you never saw, someone you never knew was the only someone for you?","title":"Sleepless in Seattle","released":1993}

{"tagline":"It's a hell of a thing, killing a man","title":"Unforgiven","released":1992}

{"tagline":"He didn't want law. He wanted justice.","title":"Hoffa","released":1992}

{"tagline":"Once in a lifetime you get a chance to do something different.","title":"A League of Their Own","released":1992}

{"tagline":"In the heart of the nation's capital, in a courthouse of the U.S. government, one man will stop at nothing to keep his honor, and one will stop at nothing to find the truth.","title":"A Few Good Men","released":1992}

{"tagline":"What if someone you never met, someone you never saw, someone you never knew was the only someone for you?","title":"Sleepless in Seattle","released":1993}

{"tagline":"It's a hell of a thing, killing a man","title":"Unforgiven","released":1992}

{"tagline":"He didn't want law. He wanted justice.","title":"Hoffa","released":1992}

{"tagline":"Once in a lifetime you get a chance to do something different.","title":"A League of Their Own","released":1992}

Figura 4.10: Nodos película que se hayan realizado entre los años 1990 y 1995

4.1.2.3. Consultas complejas para evaluaciones complejas

Para estas consultas tendremos un fin en concreto para hacer futuras relaciones.

- Encontrar actores con los que cierto actor aún no haya trabajado, pero sus co-actores sí. En la Figura 4.11 vemos como buscar a los coactores que no trabajaron con el actor Tom Cruise.

```
MATCH (a:Person {name:'Tom Cruise'})-[:ACTED_IN]→(m)←[:ACTED_IN]-(coActors),
      (coActors)-[:ACTED_IN]→(m2)←[:ACTED_IN]-(cocoActors)
WHERE NOT (a)-[:ACTED_IN]→()←[:ACTED_IN]-(cocoActors) AND a <> cocoActors
RETURN cocoActors.name AS Recommended, count(*) AS Strength ORDER BY Strength DESC
```

"Recommended"	"Strength"
"Tom Hanks"	35
"Greg Kinnear"	28
"Helen Hunt"	21
"Danny DeVito"	21
"River Phoenix"	14
"Marshall Bell"	14
"Wil Wheaton"	14
"John Cusack"	14
"Corey Feldman"	14
"Sam Rockwell"	14
"Gary Sinise"	14

Figura 4.11: Tablero en orden descendente de la cantidad de coactores de actores principales

- Encontrar todas las personas que participaron en una película en concreto y que roles tuvieron en dicha película. En la Figura 4.12 vemos como retornar todas las personas relacionadas a una película en concreto.

```
MATCH (people:Person)-[relatedTo]-(:Movie {title:'Cloud Atlas'})
RETURN people.name, type(relatedTo), relatedTo
```

"people.name"	"type(relatedTo)"	"relatedTo"
"Tom Hanks"	"ACTED_IN"	{"roles":["Zachry","Dr. Henry Goose","Isaac Sachs","Dermot Hoggins"]}
"Jim Broadbent"	"ACTED_IN"	{"roles":["Vyvyan Ayrs","Captain Molyneux","Timothy Cavendish"]}
"Tom Tykwer"	"DIRECTED"	{}
"Lana Wachowski"	"DIRECTED"	{}
"Jessica Thompson"	"REVIEWED"	{"summary":"An amazing journey","rating":95}
"Halle Berry"	"ACTED_IN"	{"roles":["Luisa Rey","Jocasta Ayrs","Ovid","Meronym"]}
"Hugo Weaving"	"ACTED_IN"	{"roles":["Bill Smoke","Haskell Moore","Tadeusz Kesselring","Nurse Noakes","Boardman Mephi","Old Georgie"]}
"Andy Wachowski"	"DIRECTED"	{}
"Jessica Thompson"	"REVIEWED"	{"summary":"An amazing journey","rating":95}
"Hugo Weaving"	"ACTED_IN"	{"roles":["Bill Smoke","Haskell Moore","Tadeusz Kesselring","Nurse Noakes","Boardman Mephi","Old Georgie"]}

Figura 4.12: Personas que interacturan en dicha película basada en las relaciones que tienen con esta.

4.2. Base de datos de investigación universitaria en Neo4j

Para los parámetros de evaluación de comparación no se tomará en cuenta el rendimiento ni la velocidad de ejecución de las consultas de dicho lenguaje, ya que, no se asegura saber que esten ejecutandose en las mismas condiciones. Sin embargo, tomaremos en cuenta los siguientes aspectos:

- Costo de la infrestructura.
- Velocidad de sintaxis (escritura y digitación de la misma).
- Comprensión del lenguaje.
- Facilidad de aprendizaje.
- Apoyo visual que sea atractivo, entendible y fácil de interpretar.

4.2.1. Comparación SQL vs CYPHER Caso 1:

La consulta de grafos escrita en Cypher es eficiente y simple. Podría decirse que puede ser formulado y realizado por trabajadores del conocimiento con menos experiencia que recibieron

la formación previa y equipado con información tal como nodo y relación de nodo. Además, la información visual que se muestra también es informativa y adecuada para diferentes niveles de espectadores, desde el conocimiento de trabajadores a ejecutivos.

En la Figura 4.13 vemos la primera consulta en lenguaje cypher el cual consta de 1 linea de código cypher el cual realiza una unión entre resultados de búsqueda especificadas por las variables respectivas encontrando los artículos, autores y co-autores del proyecto de investigación buscado , muestra quienes interactuan de manera relacional con dicho proyecto de investigación. Nos da la facilidad como desarrolladores o analistas de ver la cantidad de personas con sus respectivos roles y una imágen gráfica el cual representa la estructura de nodos y relaciones que involucran a dicho proyecto de investigación.

Una consulta más larga para responder a la misma pregunta en un relacional La base de datos sería similar a la de la Figura 4.14. La diferencia es clara. En una base de datos relacional, el requisito debe ser muy explícito por adelantado para que se traduzca correcta y cuidadosamente en consultas antes de decidir qué tablas unir en qué columnas y qué campos deben recuperarse. De hecho, requiere algunos nivel de experiencia técnica para ejecutarlo.

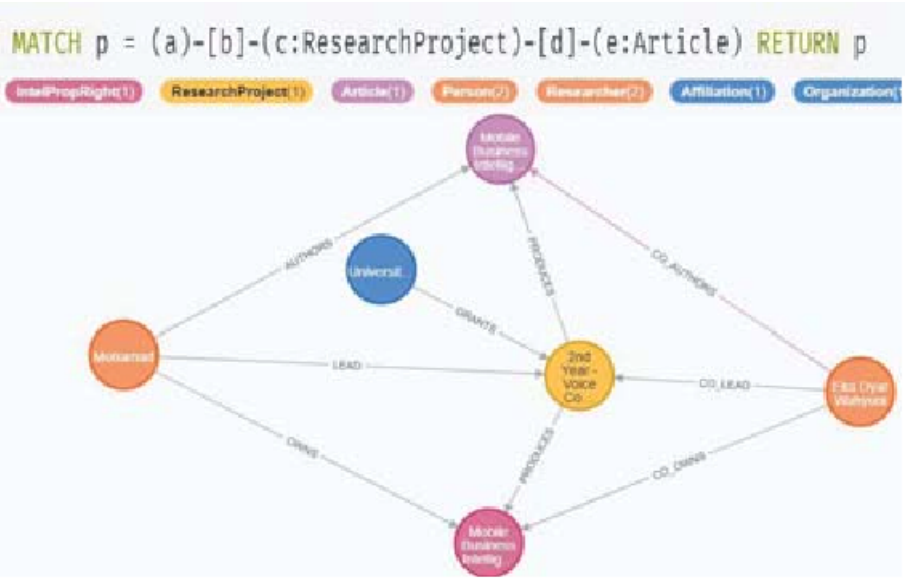

Figura 4.13: Caso 1: consulta Cypher

```
SELECT  a.name,
        rp.title,
        p.name,
        rpm.role,
        ar.title,
        aa.role
FROM  affiliation a
JOIN research_project rp on (rp.id_affiliation = a.id)
JOIN research_project_member rpm (rp.id_project =
rpm.id_project)
JOIN article ar on (ar.id_project = rp.id_project)
JOIN article_authors aa on (ar.id_art = aa.id_art)
JOIN person p on (p.id_person = aa.id_person) and
(p.id_person = rpm.id_person)
```

Figura 4.14: Caso 1 : Consulta SQL

+

4.2.2. Comparación SQL vs CYPHER Caso 2:

La consulta Cypher en la Figura 5.1 es abstracta y más natural ya que las relaciones y los nodos subyacentes no son claramente definido, pero solo se supone que existe. Sin embargo,los gráficos devueltos son informativos y brindan rápidamente información útil, que en una base de datos relacional es difícil de proporcionar. Usando una base de datos relacional como se muestra en la Figura 5.2, la consulta requiere pasos más largos, uniones y tiempo para ejecutarlo, especialmente cuando el tamaño de la base de datos es grande. Como la consulta anterior, es demasiado técnico y no fácilmente entendido por la gente en general y incluso trabajadores del conocimiento.

```
MATCH (a:Researcher)-[c:AUTHORS]-(d)-[e]-(b:Researcher)
WHERE a.name = 'Jariyah'
WITH a,b,c,d,e
MATCH (d)-[g]-(f:Pubname) return a,b,c,d,e,f,g
```

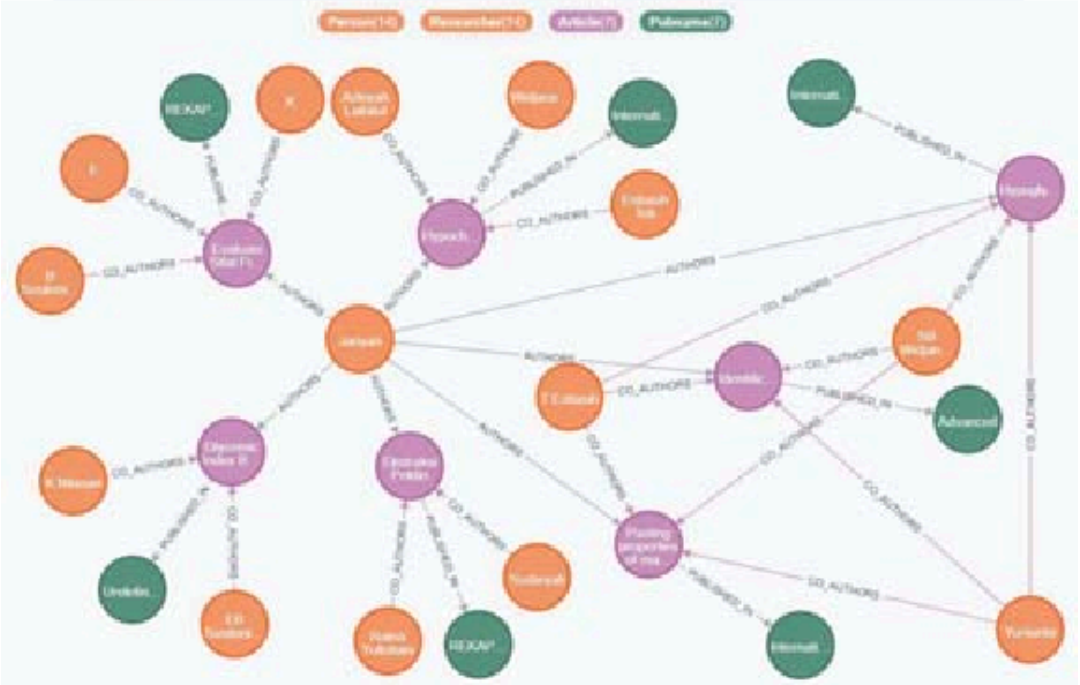

Figura 4.15: Caso 2 : Consulta Cypher

```
SELECT  pn.name,
        ar.title,
        p.name,
        aa.role
FROM pubname pn
JOIN article ar on (ar.id_pubname = pn.id_pubname)
JOIN article_authors aa on (ar.id_art = aa.id_art)
JOIN person p on (p.id_person = aa.id_person)
WHERE ar.id_art in
(
    SELECT arta.id_art
    FROM article_authors arta
    JOIN person per ON (p.id_person = arta.id_person)
)
```

Figura 4.16: Caso 3 : Consulta SQL

4.2.3. Comparación SQL vs CYPHER Caso 3:

Para el caso 3, la pregunta espera que exista(n) relación(es) entre organizaciones de una forma u otra, ya sea directamente o indirectamente. Una relación directa (a un enlace de distancia) sería algo así como Organización X HAS_MOU_WITH como Organización Y. Por otro lado, una conexión indirecta probablemente se manifestaría en algo como la Organización X EMPLEA a una Persona P que ESCRIBE un Artículo que es CO_ESCRITO por una Persona P2 que TRABAJA para la Organización Y. En este caso, hay cuatro relaciones entre Organización X y Organización Y. La respuesta en CYPHER es tan natural como puede ser, como se muestra en la Figura 4.17. Como se demostró en el caso 1, caso 2 y caso 3, el gráfico la base de datos y la consulta hacen el trabajo de responder consultas de manera eficiente en comparación con la base de datos relacional. Estos simples experimentos están en línea con estudios previos que compararon base de datos de gráficos (neo4j) y otros sistemas de bases de datos. Los resultados también revelan que las consultas gráficas formuladas están muy cerca de los lenguajes naturales indicados en las consultas de perspectivas múltiples. Para directores o líderes universitarios que a menudo se encuentran en medio de reuniones en las que se les solicita para proporcionar información aleatoria y ad-hoc sobre su universidades, las capacidades de las consultas gráficas sin duda serán beneficioso. Además de las funciones de informes y análisis compatibles por consultas y bases de datos relacionales, que la mayoría de los ejecutivos Los sistemas de información (EIS) que tienen actualmente, las organizaciones pueden necesitar desarrollar capacidades gráficas y analíticas respaldadas por bases de datos gráficas y consultas. Estudios previos por capacidades de chatbot agregadas a la

EIS existente para que los ejecutivos puedan ver y obtener información sobre sus universidades ordenando a través de voces y textos. Los problemas con esos sistemas son que no se sabe más allá del conocimiento específico incrustado en ellos. Los comandos son limitados porque ellos y sus consultas SQL asociadas deben estar predefinidos primero. Entonces, esos sistemas tienen solicitudes de información aleatorias y ad hoc no atendidas. La base de datos de gráficos y las consultas tienen grandes posibilidades de superar ese problema. Pueden entender y producir los resultados de manera intuitiva a pesar de las preguntas a ciegas, que parecen difíciles de comprender y parecer sin sentido, como se muestra en el Caso 1 y Caso 2. El nodo-relación-nodo posiblemente se convertirá en el clave en la interpretación de solicitudes aleatorias de información siempre que podemos mapear cosas a nodos en la base de datos. De hecho, la implementación seguirá siendo un desafío, especialmente cuando se integran datos de diferentes bases de datos, a menudo sistemas heredados, donde la documentación está menos disponible o no existe.

```
MATCH  p = (a:Affiliation)-[*1..4]-(i:Affiliation)
WHERE  (lower(a.name) contains 'jawa timur'
   OR   lower(a.name) contains 'jatim')
   AND (lower(i.name) contains 'nopember'
   OR   lower(i.name) contains 'airlangga')
   OR   lower(i.name) contains 'diponegoro'
RETURN p
LIMIT 20
```

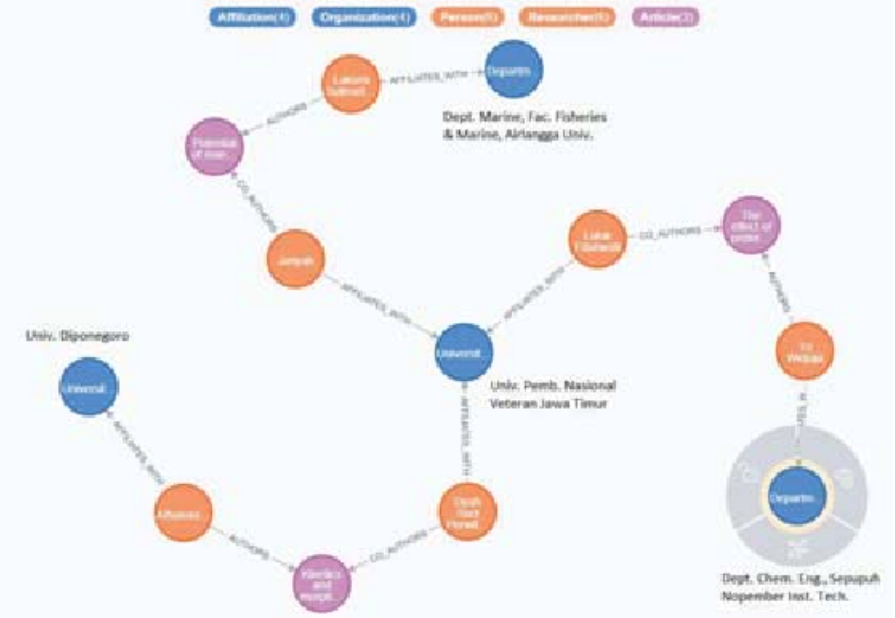

Figura 4.17: Caso 3 : Consulta Cypher

4.2.4. Ventajas que propone el Neo4j con la base de datos en lenguaje Cypher

Al basarnos en un sistema en concreto, trabajando con una base de datos de investigadores para el respaldo de la información de los estudiantes universitarios, se detectó las siguientes ventajas:

- La base de datos de gráficos de investigación presentada en este estudio tiene demostrado responder de manera eficiente a las consultas en comparación con la base de datos relacional.
- La base de datos de gráficos de investigación presentada en este estudio proporcionan visualmente atractivos discernimiento.
- La sintaxis del lengua Cypher se asemeja al lenguaje natural para la correcta interpretación

de una persona con un conocimiento promedio de dicha disciplina, apoyado de un material visual que respalda dicha sintaxis.

- Sql es destacado por ser un lenguaje usado netamente por programadores de mucha experiencia, los cuales tienen mucho tiempo estudiando no solo el entorno de bases de datos sino también el lenguaje de SQL, como realizar transacciones y consultas correspondientes, sin embargo, se apoyan de herramientas como la ingeniería reversa para plasmar sus arquitecturas en tablas organizacionales las cuales muestran de manera robótica los nodos y relaciones.

- Las consultas SQL extensas no solo requieren más tiempo para ejecutarse, sino que también es más probable que incluyan errores de codificación humana debido a su complejidad. Además, las consultas más breves aumentan la facilidad de comprensión y mantenimiento en todo su equipo de desarrolladores.

- Cypher está diseñado para que los desarrolladores, los profesionales de bases de datos y las partes interesadas del negocio lo lean y entiendan fácilmente. Es fácil de usar porque coincide con la forma en que describimos intuitivamente los gráficos mediante diagramas.

- La noción básica de Cypher es que le permite pedirle a la base de datos que busque datos que coincidan con un patrón específico. Coloquialmente, podríamos pedirle a la base de datos que “encuentre cosas como esta”, y la forma en que describimos cómo se ven “cosas como esta” es dibujándolas usando arte ASCII .

- Un lenguaje de consulta representa su modelo de cerca. Es por eso que SQL se trata de tablas y JOIN, mientras que Cypher se trata de relaciones entre entidades. Tanto como el modelo gráfico es más natural para trabajar, también lo es Cypher, ya que toma prestado de la representación pictórica de círculos conectados con flechas que cualquier interesado (ya sea técnico o no técnico) puede entender.

- En una base de datos relacional, el proceso de modelado de datos se abstrae hasta ahora de las consultas SQL reales del día a día que existe una gran disparidad entre el análisis y la implementación. En otras palabras, el proceso de construcción de un modelo de base de datos relacional no es adecuado para hacer (y responder) preguntas de manera eficiente desde ese mismo modelo. Los modelos de bases de datos de gráficos, por otro lado, no solo comunican cómo se relacionan sus datos, sino que también lo ayudan a comunicar claramente los tipos

de preguntas que desea hacer sobre su modelo de datos. Los modelos gráficos y las consultas gráficas son solo dos caras de la misma moneda.

Capítulo 5

Resultados y/o Discusión

5.0.1. Neo4j como la mejor opción de BD

Después de haber reproducido todas las técnicas correspondientes, notamos que Neo4j es mejor que SQL tanto en sintaxis como en aprendizaje para programadores que recien se adentran a esta rama,

Neo4j no solo te brinda una sintaxis mas entendible y reducida, sino también te brinda una gráfica que te permite entender mejor la consulta y relaciones de tu base de datos sin estar realizando comandos como select, join y demás para mostrar lo que deseas.

Los colores interactivos que brinda su gráfica hace el trabajo de análisis y comprensión menos tediosa.

En la comparación vemos que 5.1 y 5.2 presentan muchas diferencias las cuales ya analizamos con anterioridad.

```
MATCH (a:Researcher)-[c:AUTHORS]-(d)-[e]-(b:Researcher)
WHERE a.name = 'Jariyah'
WITH a,b,c,d,e
MATCH (d)-[g]-(f:Pubname) return a,b,c,d,e,f,g
```

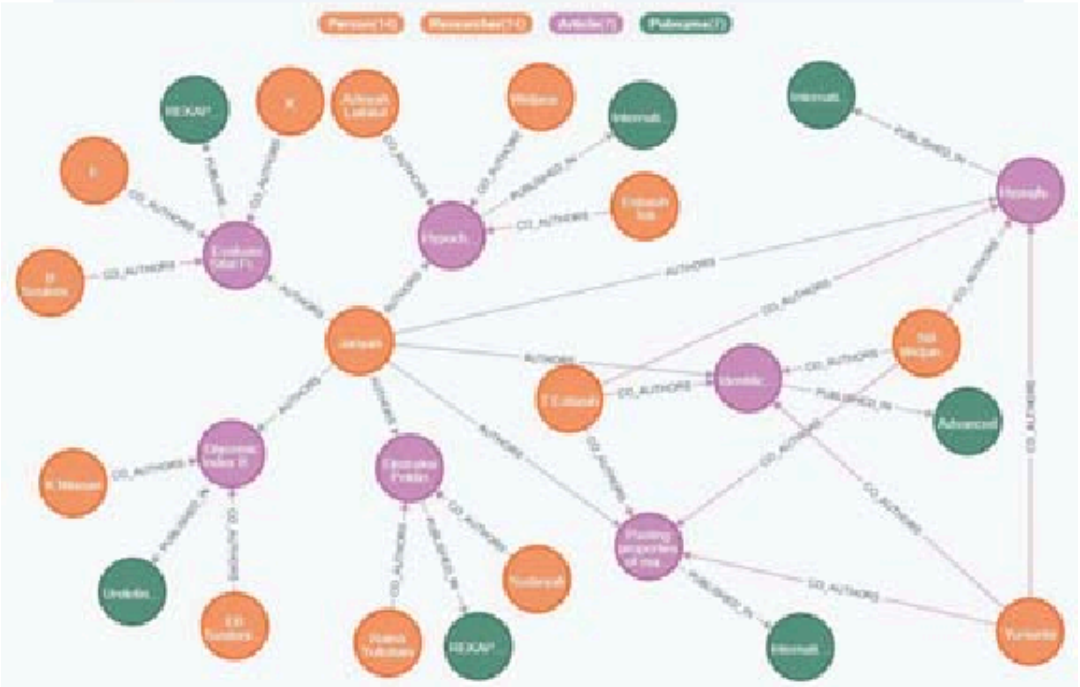

Figura 5.1: Caso 2 : Consulta Cypher

```
SELECT   pn.name,
         ar.title,
         p.name,
         aa.role
FROM pubname pn
JOIN article ar on (ar.id_pubname = pn.id_pubname)
JOIN article_authors aa on (ar.id_art = aa.id_art)
JOIN person p on (p.id_person = aa.id_person)
WHERE ar.id_art in
(
    SELECT arta.id_art
    FROM article_authors arta
    JOIN person per ON (p.id_person = arta.id_person)
)
```

Figura 5.2: Caso 3 : Consulta SQL

5.0.2. Neo4j como la mejor opción para analístas

La herramienta propuesta no solo brinda una gráfica iteractiva sino que además muestra las relaciones que posiblemente puedan llevarte a un objetivo que estes buscando.
Muestra toda la información correspondiente de cada nodo con su respectiva relación de la cual quisieras información y los posibles nodos que se vieran involucrados.

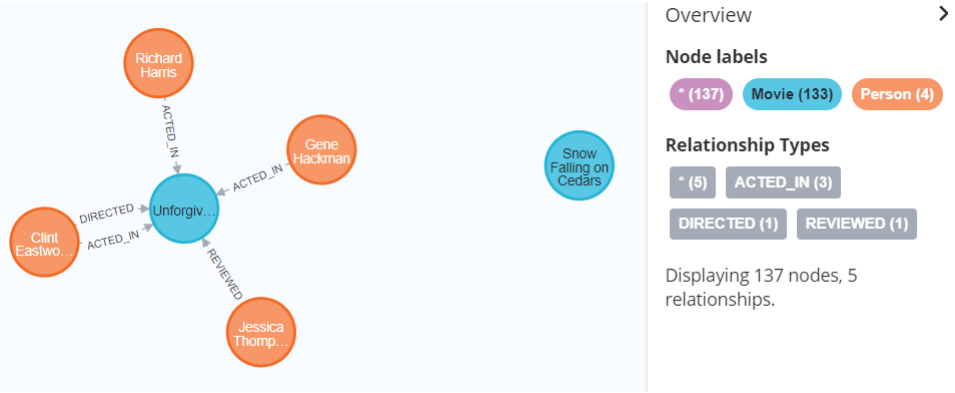

Figura 5.3: Nodo película Unforgiven con sus respectivas relaciones

5.0.3. Concytec como Información Valiosa

En la técnica 2 pudimos evaluar como usan google schoolar como base de datos de investigadores para el mismo propósito de este papper, hacer que los universitarios tengan el mayor respaldo y

asesoramiento posible de la manera mas accesible que puedan encontrar.

Concytec asi como google schoolar, brindan cierta información valiosa para un universitario el cual le permitirá elegir mejor a su asesor de acuerdo a lo que requieran. La segunda técnica mostró que es una necesidad realizar este tipo de propuestas para hacer que grupos de investigación como el que tiene concytec puedan ayudar a los demás.

Figura 5.4: CV de profesor en la página de concytec

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

Al desarrollar este trabajo de investigación y apoyarnos de dos técnicas tanto informativas e introductorias como comparativas para nuestro análisis y resultado obtenido, notamos que, fue una elección correcta el elegir a Cypher como lenguaje de programación y a Neo4j como herramienta para nuestra investigación.
De entre tantas fuentes de información es importante elegir una de las más accesibles para el investigador, en este caso, Concytec. Al evaluar dichas fuentes de información similares a esta, vimos que , el avance de investigación para el reforzamiento de asesorias y respaldo para los investigadores es impresindible para lograr una mejor y más asertiva búsqueda con las herramientas correctas,

6.2. Trabajos futuros

Se plantea como trabajo futuro realizar una aplicación para la página de concytec el cual recopile la información de los CV de los investigadores y así hacer el llenado de nuestra base de datos orientada a grafos de manera automática, el cual facilitaría la recopilación de datos y podría ampliar nuestro avance de investigación planteado en este artículo.

Bibliografía

[Afandi and Wahyuni, 2020] Afandi, M. I. and Wahyuni, E. D. (2020). University research graph database for efficient multi-perspective data analysis using neo4j. In *2020 6th Information Technology International Seminar (ITIS)*, pages 286–290. IEEE.

[Agudo Merino, 2020] Agudo Merino, A. (2020). Base de datos distribuida de grafos con neo4j para el record linkage de redes sociales.

[Aryani et al., 2017] Aryani, A., Wang, J., Zhang, H., Xiang, A., Zhou, Z., and Wang, K. (2017). Visualising research graph using neo4j and gephi.

[Fernandes and Bernardino, 2018] Fernandes, D. and Bernardino, J. (2018). Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In *Data*, pages 373–380.

[Gehani, 1990] Gehani, N. (1990). Introduction to database systems.

[Guia et al., 2017] Guia, J., Soares, V. G., and Bernardino, J. (2017). Graph databases: Neo4j analysis. In *ICEIS (1)*, pages 351–356.

[Holzschuher and Peinl, 2013] Holzschuher, F. and Peinl, R. (2013). Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 195–204.

[Lu et al., 2017] Lu, H., Hong, Z., and Shi, M. (2017). Analysis of film data based on neo4j. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 675–677. IEEE.

[Miller, 2013] Miller, J. J. (2013). Graph database applications and concepts with neo4j. In *Proceedings of the southern association for information systems conference, Atlanta, GA, USA*, volume 2324.

[Stothers and Nguyen, 2020] Stothers, J. A. and Nguyen, A. (2020). Can neo4j replace postgresql in healthcare? *AMIA Summits on Translational Science Proceedings*, 2020:646.

[Vukotic et al., 2015] Vukotic, A., Watt, N., Abedrabbo, T., Fox, D., and Partner, J. (2015). *Neo4j in action*, volume 22. Manning Shelter Island.