

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/318416546>

Graph database: A case study for detecting fraud in acquisition of Brazilian Government

Conference Paper · June 2017

DOI: 10.23919/CISTI.2017.7975974

CITATIONS

5

READS

321

4 authors, including:



Gustavo Cordeiro Galvão van Erven

University of Brasília

14 PUBLICATIONS 294 CITATIONS

[SEE PROFILE](#)



Maristela Holanda

University of Brasília

186 PUBLICATIONS 944 CITATIONS

[SEE PROFILE](#)



Célia Ghedini Ralha

University of Brasília

152 PUBLICATIONS 721 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Database Administrators [View project](#)



Women in Computing [View project](#)

Banco de Dados em Grafo: Um Estudo de Caso em Detecção de Fraudes no Governo Brasileiro

Graph Database: A Case Study for Detecting Fraud in acquisition of Brazilian Government

Nomes de Autores da 1ª Instituição

Linha 1 (da Instituição) Depart., Organização, Universidade
Linha 2 (da Instituição)
Linha 3: Cidade, País
Linha 4: Endereços de Email

Resumo — As licitações consistem em uma das principais formas de aquisição do Governo Federal e estão sujeitas a fraudes e corrupção. Dentro desse contexto, diversas instituições, como a Controladoria-Geral da União (CGU), buscam identificar indícios de fraudes, como consultas em bancos de dados por relacionamentos entre empresas participantes de um mesmo processo de compra. No entanto, muitas dessas consultas são realizadas por meio manual ou limitadas nos sistemas de informação por razões da arquitetura de armazenamento de dados atual. Este artigo explora vantagens de utilizar bancos NoSQL de grafos na identificação de relacionamentos entre empresas para detecção de indícios de fraudes em licitações. Neste contexto, foi definido um estudo de caso utilizando informações de sócios e empresas e realizadas consultas de interesse dos auditores da CGU.

Palavras Chave - Banco de Grafos; Detecção de Fraudes; NoSQL; Banco Relacional.

Abstract — Procurement is one of the main forms of acquisition of the Federal Government and are subject to fraud and corruption. In this context, many institutions such as the Brazilian Office of the Controller General try to identify fraud evidences, e.g., use of queries in databases with information about partnership of companies owned by the same group of people to simulate competition. However, many of these queries are manual or limited by the use of information systems databases. This paper explore the benefits of using graph NoSQL databases to identify the relationships between companies to detect fraud in procurement. A case study was carried out to validate the use of NoSQL databases using companies' partnership information based on queries defined by the CGU's auditors.

Keywords - Graph Database; Fraud Detection; NoSQL; Relational Database.

I. INTRODUÇÃO

Os pregões são utilizados para aquisições de bens e serviços pela Administração Pública Federal (APF) e são

Nomes de Autores da 2ª Instituição

Linha 1 (da Instituição) Depart., Organização, Universidade
Linha 2 (da Instituição)
Linha 3: Cidade, País
Linha 4: Endereços de Email

normatizados na Lei 10.520/2002¹. Os pregões funcionam como um leilão invertido, onde os interessados em vender ou prestar um serviço para a APF dão lances de menor valor para oferecer o item ou itens descritos no processo de compra. No entanto, esse processo de licitação está exposto a diversos tipos de fraudes, como simulação de concorrência pelas participantes.

Entre os órgãos que combatem esse tipo de fraude está a Controladoria-Geral da União² (CGU) que realiza auditorias e fiscalizações nas entidades da APF. Como um dos trabalhos de fiscalização da CGU pode-se citar a detecção de relacionamentos suspeitos que possam implicar em algum tipo de fraude. Esse processo de detecção consiste na busca em bancos de dados, pelos auditores, de relacionamentos entre pessoas e empresas, como relações societárias, parentescos, endereços, dentre outros. Entretanto, esse processo é realizado por meio de consultas (*queries*) pontuais em banco de dados relacionais, construídas muitas vezes pelo próprio auditor, geralmente muito complexas sendo compostas de várias junções entre tabelas. Devido ao impacto de desempenho dessas consultas, muitas vezes, os sistemas utilizados atualmente limitam a pesquisa em apenas três relacionamentos.

Nesse contexto, os bancos de grafos se apresentam como uma abordagem útil para se utilizar no armazenamento e recuperação dos dados dessa natureza, por serem os relacionamentos facilmente representáveis com grafos. Essa estrutura é utilizada em diversas aplicações, como em redes de relacionamento [13], sendo muito utilizada em investigações, como fraudes e outras ações criminais [8, 11, 15]. Sendo assim, o uso de grafos permite elaborar sistemas para identificar relacionamentos entre entidades, como pessoas e empresas, evidenciando conluios entre grupos que se formam para tirar proveito ilícito de alguma situação, sendo uma destas a fraude em pregões do Governo Federal.

¹ http://www.planalto.gov.br/ccivil/_03/leis/2002/L10520.htm

² <http://www.cgu.gov.br>

Dessa forma, este artigo explora benefícios da tecnologia de banco de grafos, em contra ponto à estrutura existente (bancos relacionais), subsidiando a decisão sobre seu uso não apenas para CGU, como também para outras instituições que realizem um trabalho similar. Para a avaliação, foram utilizados os bancos MySQL³ (relacional) e o Neo4j⁴ (grafo). Inicialmente foram feitos experimentos sobre o impacto da mudança na estrutura das consultas, em particular avaliando a simplicidade da sua construção, e em seguida foram avaliadas quais vantagens seu uso poderia trazer para o desempenho do sistema. Como estudo de caso foi selecionada a busca de vínculos por sociedades em uma profundidade acima do que o sistema já realiza (três níveis) e definida pelos auditores, o que permitiria aumentar as chances de encontrar um relacionamento de interesse.

O restante deste artigo se apresenta da seguinte forma: na Seção II são apresentados os bancos de grafos e trabalhos relacionados; na Seção III é apresentada a metodologia utilizada no estudo; na Seção IV são descritos o modelo relacional e o modelo conceitual de grafos e a análise das consultas; na Seção V as conclusões e trabalhos futuros são apresentados.

II. BANCO DE DADOS EM GRAFOS

Os chamados bancos NoSQL, derivados dos trabalhos da Amazon [5] e Google [3], vêm ganhando espaço como novas propostas de modelos e sistemas de bancos de dados para o gerenciamento de informações. Pela sua crescente importância, essas novas formas de armazenamento e manipulação de informação já vêm sendo tema de diversos trabalhos [4, 7, 9, 10, 14]. Entre as tecnologias de NoSQL estão os bancos de grafos, os quais utilizam a estrutura de grafo para organizar os dados, permitindo uma melhor aderência em aplicações onde os relacionamentos entre entidades são tão importantes quanto seus atributos [13].

Os bancos NoSQL de grafos podem ser caracterizados pela natureza do grafo da seguinte forma [1]:

- *Grafo simples*: os vértices e arestas possuem apenas rótulos ou um valor;
- *Hipergrafo*: grafos que permitem que suas arestas possuam mais de dois vértices;
- *Grafo com atributos*: grafos que permitem que seus vértices e arestas tenham atributos;
- *Grafo aninhado e hipernó*: grafos que suportam outros grafos como nós, ou seja, todo um grafo pode ser vinculado com outro, como se fosse um vértice.

Para esse trabalho foi utilizado um grafo com atributos pelo fato deste possuir uma maior flexibilidade na modelagem. Esse tipo de grafo também foi utilizado em outros trabalhos [2, 12] mostrando as vantagens na travessia das instâncias no banco de grafos (percorrer os vértices), pela diminuição do número de junções.

III. METODOLOGIA

A metodologia deste trabalho é apresentada na Figura 1 e foi dividida em duas fases, sendo a primeira com o objetivo de criar o ambiente de avaliação e executar a consulta definida pelos auditores da CGU, permitindo a análise da estrutura em ambas as tecnologias. Na segunda fase foram realizados testes de desempenho e uso de uma função do banco de grafos para resolver a busca de menor caminho.

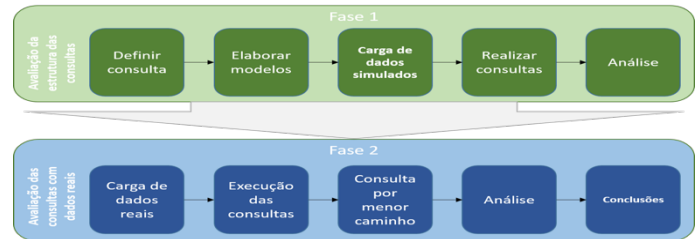


Figura 1: Metodologia.

A. Fase I

Nessa fase, foi definida a consulta e o modelo em ambas as tecnologias para executá-la. Um volume pequeno de dados simulados foi utilizado, o que permitiu a análise no uso dos bancos sem preocupação com o desempenho, assim como evitar problemas de sigilo das informações.

1) *Definir Consulta*: Para avaliar o uso da abordagem relacional e de grafos, foi escolhida uma consulta de importância para os auditores da CGU: **Como duas empresas se relacionam a partir de suas sociedades até o terceiro nível de empresa?** Ela procura identificar relacionamentos entre os sócios a partir de empresas e sociedades intermediárias entre duas concorrentes em uma licitação. O terceiro nível de empresa implica em uma profundidade de até seis níveis, como apresentado na Figura 2.

Esse vínculo foi escolhido considerando que, dado um conjunto de empresas em uma licitação, o ideal é não existir relacionamentos diretos ou indiretos entre elas. Caso estes existam, pode-se caracterizar um indício de conluio, como por exemplo, empresas que possuem parentes como sócios. Pessoas jurídicas também podem ser sócias de outras empresas, o que significa que o vínculo de sociedade não é apenas com pessoas físicas. Isso pode ser utilizado para distanciar um relacionamento e tornar a fraude mais difícil de ser percebida.

A partir da definição dessa consulta, foi selecionado o relacionamento entre sócios e empresas para construir um Banco de Relacionamentos de Sociedades.

2) *Elaborar modelo*: Foi criado inicialmente um modelo relacional descrevendo o relacionamento entre os sócios de empresas e em seguida elaborado o modelo de dados para o banco de grafos. O objetivo foi modelá-lo de forma simples, mantendo as informações mais relevantes inicialmente, como quem é sócio de qual empresa e sua qualificação da sociedade. Outra característica das sociedades incluída foi a qualificação do sócio. Cada sócio pode desempenhar papéis diferentes, como cotista ou diretor, que podem ser importantes para o auditor no momento da análise do relacionamento, como

³ <http://www.mysql.com/>

⁴ <http://www.neo4j.com/>

3) *Carga de dados simulados:* Para popular os bancos, foram inseridos dados fictícios de pessoas e empresas, assim como gerados os vínculos de forma aleatória no banco

2) *Execução das consultas:* As consultas foram realizadas no mesmo servidor de forma alternada em cada banco, parando-se o serviço do banco ocioso para evitar interferências.

```

graph TD
    A([Empresa I]) --> B([Pessoa])
    A --> C([Empresa (alvo?)])
    B --> D([Empresa (alvo?)])
    C --> E([Pessoa])
    C --> F([Empresa (alvo?)])
    D --> G([Pessoa])
    D --> H([Empresa (alvo?)])
    E --> I([Empresa (alvo?)])
    E --> J([Pessoa])
    F --> K([Pessoa])
    F --> L([Empresa (alvo?)])
    G --> M([Pessoa])
    G --> N([Empresa (alvo?)])
    H --> O([Pessoa])
    H --> P([Empresa (alvo?)])
    I --> Q([Pessoa])
    I --> R([Empresa (alvo?)])
    J --> S([Pessoa])
    J --> T([Empresa (alvo?)])
    K --> U([Pessoa])
    K --> V([Empresa (alvo?)])
    L --> W([Pessoa])
    L --> X([Empresa (alvo?)])
    M --> Y([Pessoa])
    M --> Z([Empresa (alvo?)])
    N --> AA([Pessoa])
    N --> AB([Empresa (alvo?)])
    O --> AC([Pessoa])
    O --> AD([Empresa (alvo?)])
    P --> AE([Pessoa])
    P --> AF([Empresa (alvo?)])

```

relacional. Posteriormente os dados foram migrados para o banco de grafos. A carga no banco de grafos foi feita exportando-se os dados em arquivos CSV (*comma-separated values*) do banco MySQL e importando-os por meio de um comando de carga do próprio Neo4j.

5) *Análise:* No fim da Fase 1 foi realizada uma análise das diferenças entre a estrutura das consultas e apresentadas as considerações sobre vantagens na mudança de tecnologia. Em seguida, foi realizada a Fase 2 para verificar benefícios que o uso dos bancos de grafos poderiam trazer para a consulta.

Na Fase 2 foi criado um ambiente com os mesmos modelos de dados da Fase 1 para execução das consultas com dados reais. O objetivo foi avaliar o desempenho sobre uma massa de dados real, assim como o uso de uma função específica do banco de grafos que retorne apenas os vínculos de menor distância (*shortest paths*).

consultas definidas na Fase 1 sobre os dados reais, foi executada uma busca no banco de grafos utilizando-se a função de menor caminho.

4) *Análise:* Com os resultados das consultas foram analisados os dados de desempenho e quais benefícios poderiam ser obtidos de ambas as tecnologias.

5) **Conclusões:** Por fim, foi concluída a Fase 2 com as conclusões finais, sintetizando os resultados de ambas as fases sobre as vantagens do uso de bancos de grafos na consulta definida pelos auditores.

Nessa seção são apresentados a descrição do modelo relacional e de grafos do Banco de Relacionamentos de Sociedades, assim como as consultas e seus resultados de execução com os dados simulados e com os dados reais.

A Fig. 3 apresenta o modelo relacional para o Banco de Relacionamentos de Sociedades (BRS) utilizado neste artigo, com as informações sobre empresas e sócios. O diagrama da Fig. 3 apresenta as informações simplificadas de sociedade, incluindo as entidades *Pessoa*, *PessoaSocio*, *Empresa*, *EmpresaSocio* e *QualificacaoSocio*. No Brasil, uma *Empresa* pode ter vários sócios, como uma *Pessoa* ou mesmo outra *Empresa*. Sendo assim, várias instancias de uma entidade pode se relacionar com diversas de outra, sendo estes

relacionamentos materializados nas tabelas associativas *PessoaSocio* e *EmpresasSocios*. Junto aos relacionamentos de sócios estão as informações de qualificação do sócio da entidade *QualificacaoSocio*. O objetivo é identificar qual o papel do sócio na empresa, tais como administrador, cotista, acionista ou diretor, por exemplo.

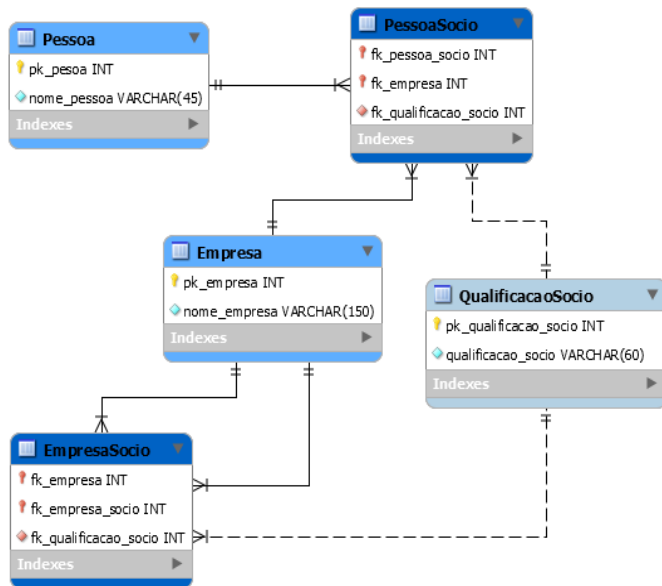


Figura 3: Modelo Relacional para vínculos de sociedades.

Para o banco de grafos foi utilizada a notação MDG-NoSQL especificada em [6] na criação de um modelo de grafos com atributos. O modelo é apresentado na Fig. 4 e estão presentes na forma de vértice as entidades *Pessoa* e *Empresa*. O relacionamento entre elas se dá como arestas do tipo *SOCIO_DE*, inclusive o auto relacionamento entre empresas.

Como o vínculo societário é de M:N, as cardinalidades são indicadas com (0..n) no modelo tanto na aresta que liga uma *Pessoa* a uma *Empresa* como sócio, como na aresta que relaciona um sócio *Empresa* com outra *Empresa*. Uma mesma instância pode apontar ou ser apontada por nenhuma ou diversas arestas, não precisando de um vértice intermediário. Essa é uma vantagem da estrutura de grafos que simplifica a representação dos relacionamentos dessa natureza.

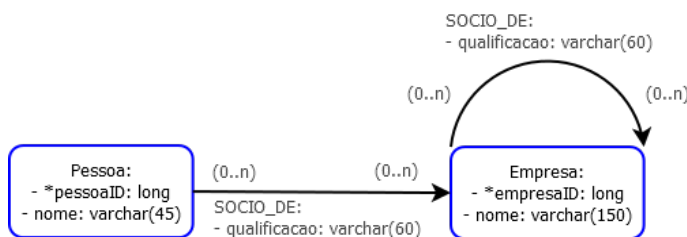


Figura 4. Modelo de grafo para vínculos de sociedade.

Outra característica é a ausência da entidade *QualificacaoSocio*, que passou a ser um atributo do vínculo de sociedade. Se esta fosse colocada como vértice, implicaria em criar um nó intermediário para associar o sócio à empresa com

determinada qualificação (uma pessoa cotista de uma empresa), o que aumentaria a normalização do modelo, mas implicaria no aumento do número de vértices e caminhos entre sócios e empresas. Para se chegar do sócio a uma empresa, com a qualificação como sócio seria necessário passar pelo nó de intermediário. Isso aumentaria em um vértice para cada vínculo buscado no caso da consulta por sociedades.

Outro impacto é a criação de vínculos entre empresas e pessoas pela qualificação. Como a qualificação estaria ligada a cada vértice intermediário, um nó da qualificação de cotista estaria criando um caminho de apenas três saltos entre uma pessoa ou empresa, mesmo que ambos nunca tivessem se encontrado. Isso porque diversas pessoas são cotistas, independente de se conhecerem em algum momento. Sendo assim, como o fato de duas pessoas terem a mesma qualificação, como dois cotistas, independente das empresas, não traz informação relevante na busca de indícios, preferiu-se colocar como um atributo do vínculo, permitindo sua filtragem e reduzindo a complexidade do modelo e o número de caminhos entre sócio e empresa, o que também favorece as consultas.

B. Realização das consultas

A consulta definida em conjunto com os auditores consiste em buscar um relacionamento entre duas empresas, imaginando que estas podem estar concorrendo em um mesmo item de licitação, por meio dos vínculos societários. O objetivo será descer até o terceiro sócio do tipo empresa, o que poderá acarretar em uma árvore de busca em até seis níveis. A Fig. 2 apresenta essa árvore, destacando o caminho de maior profundidade.

Foi realizada a consulta sobre cada banco. Inicialmente, verificou-se a complexidade para expressá-la como SQL ou na linguagem do banco de grafos Neo4j, o *Cyper*⁵. Em seguida, foram levantados alguns recursos que podem impactar no desempenho da consulta no banco de grafos, e avaliada brevemente a execução de uma delas sobre dados reais.

Para consultar as sociedades no modelo relacional foi necessário construir uma consulta para cada ramo da árvore da Fig. 2, devido a variação na passagem das entidades, como *Empresa - Pessoa* e *Empresa - Empresa*. Foram então construídas oito consultas para cobrir as diversas possibilidades de sociedades até a terceira empresa, sendo a busca do caminho mais longo apresentada parcialmente na Consulta 1 executada no MySQL. Para obter os seis níveis de sócios, foram necessárias doze junções (linha 5 a 11 da Consulta 1) alternando entre empresas e pessoas, isso para apenas um dos caminhos. Regras de exclusão para evitar caminhos repetidos também foram incluídos na cláusula *where* (linha 13 a 21 da Consulta 1), o que aumenta a complexidade da consulta. Os demais caminhos tiveram estruturas mais simples, mas ainda com diversas junções.

```
#EMP - PES - EMP - PES - EMP - PES - EMP
```

```
1 SELECT
```

⁵ <http://neo4j.com/developer/cypher-query-language/>

```

2  C01.nome_empresa, P01.nome_pessoa, ...
3  FROM PessoasSocios AS PP01
4  --Inicio das 12 juncoes.
5  INNER JOIN Empresas AS C01
6  ON C01.pk_empresa = PP01.fk_empresa
7  INNER JOIN Pessoas AS P01
8  ON P01.pk_pessoas = PP01.fk_pessoa_socio
9  ...
10 LEFT JOIN Empresas AS C04
11  ON C04.pk_empresa = PP06.fk_empresa
12 --Fim das 12 juncoes
13 WHERE C02.pk_empresa <> C01.pk_empresa
14 AND C02.pk_empresa <> C03.pk_empresa
15 ...
16 AND ((C01.nome_empresa = 'EMPRESA B'
17      AND C02.nome_empresa = 'EMPRESA C')
18 OR (C01.nome_empresa = 'EMPRESA B'
19      AND C03.nome_empresa = 'EMPRESA C')
20 OR (C01.nome_empresa = 'EMPRESA B'
21 AND C04.nome_empresa = 'EMPRESA C'));

```

Consulta 1. Busca Empresa Pessoa sociedade em SQL.

Em contraponto, para realizar a mesma consulta no banco de grafos Neo4j foi utilizada a linguagem Cypher, que permite descrever um padrão de grafo que se deseja recuperar do banco, que realiza uma busca no Neo4j por um subgrafo com um padrão que tenha os vértices da EMPRESA B e EMPRESA C ligados até seis níveis por arestas do tipo SOCIO_DE. Da forma como essa consulta foi construída, não faz diferença se os vértices intermediários são empresas ou pessoas, se eles estiverem ligados por vínculos de sociedades entre essas duas

empresas a linguagem irá incluí-los no grafo de retorno, como apresentado na Fig. 5, utilizando-se a ferramenta de visualização web disponível no Neo4j.

```

1 MATCH (empresaB { nome : 'EMPRESA B' }) -
[*:SOCIO_DE*1..6] - (empresaC { nome : 'EMPRESA C' })
RETURN empresaB, r, empresaC;

```

Consulta 2. Busca Empresa Pessoa sociedade em Cypher.

A comparação entre as Consultas 1 e 2 demonstra o poder de síntese que uma linguagem especializada para bancos de grafos possui para a travessia entre vértices na busca por relacionamentos entre entidades. Isso sem considerar a possibilidade do uso de funções existentes em alguns bancos que aplicam algoritmos típicos de grafos.

Entre essas funções está a de menor caminho entre dois vértices, para casos onde identificar apenas um vínculo já é suficiente, caso explorado na Fase 2 desse trabalho, onde foi utilizado a função *allShortestPaths* do Neo4j.

C. Breve Análise Comparativa Sobre Dados Reais

Para avaliar a vantagem de desempenho entre os bancos e o uso de uma função de menor caminho, foram carregados nos modelos dados reais de sociedades e escolhidas duas empresas com aproximadamente 400 sócios para verificar se possuíam algum relacionamento entre elas. As consultas foram realizadas em uma máquina com o sistema operacional Linux, 24 núcleos de processamento e 64 GB de RAM. A Tabela 1 apresenta os detalhes do ambiente de avaliação utilizado.

Foram carregados mais de 20 milhões de relacionamentos de sociedades em ambos os bancos, conforme a Tabela 2. Em seguida, foram executadas a consulta de maior ramo para o MySQL (versão 5.1.x) e o de menor caminho para o Neo4j (versão 2.1.x).

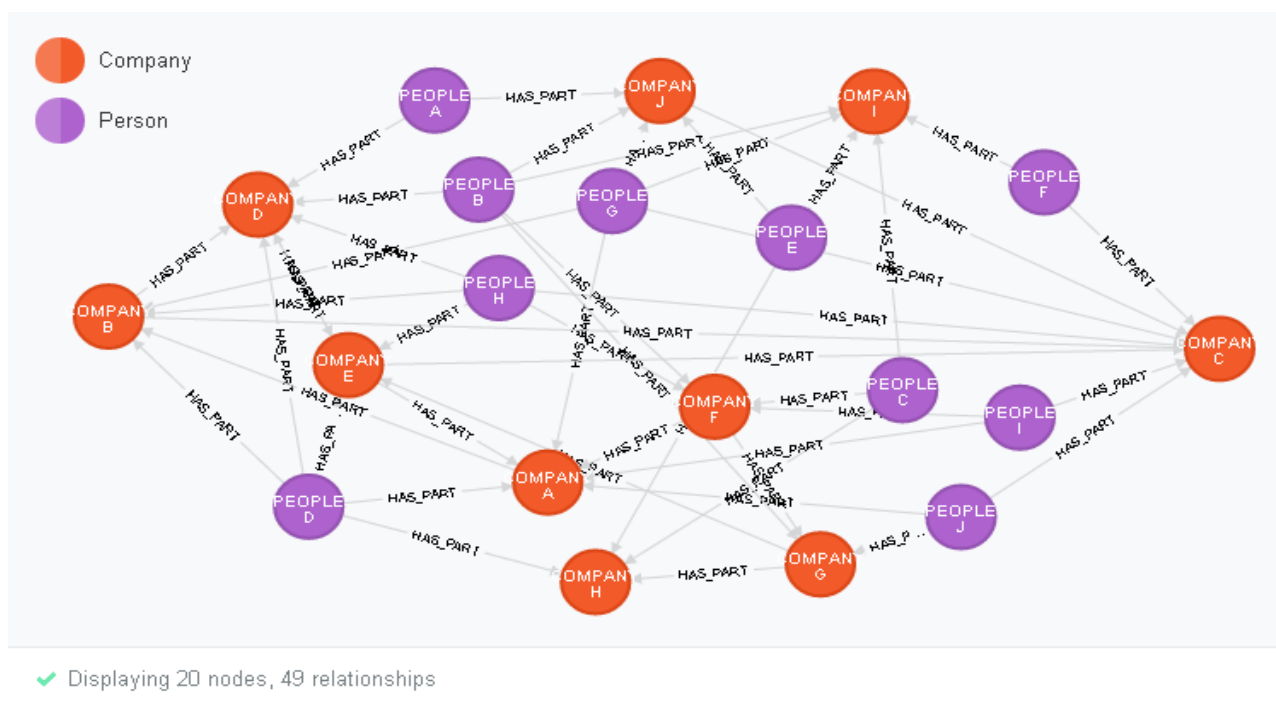


Figure 5. Resultado da consulta por sócio entre duas empresas.

TABELA I. DETALHES DO AMBIENTE DE AVALIAÇÃO.

Fonte	Base CGU
Número Empresas	7.754.989
Número Sócios	14.190.151
Número de Sociedades	20.903.480
Máquina	24 cores 64GB de RAM
Consulta	Alternada de seis níveis
Origem	Uma empresa com aproximadamente 400 sócios
Destino	Uma segunda empresa com aproximadamente 400 sócios
Objetivo	Avaliar a busca por relacionamentos

Tanto para o MySQL como para o Neo4j foram extraídas as informações de sociedade do banco de dados de origem e colocadas em tabelas temporárias de um banco intermediário. A carga no MySQL foi realizada diretamente entre as tabelas temporárias. Já para o Neo4j foi necessário exportar os dados de sociedades para um arquivo e quebrá-lo em diversos outros para permitir o carregamento.

TABELA II. COMPARAÇÃO ENTRE O MYSQL E O NEO4J.

	MySQL	Neo4j
Método de carga	Ferramenta ETL	Aplicação desenvolvida em java
Tempo de execução da consulta	72.608,949s (Aprox. 20h)	14h sem sucesso para consulta por padrões e 369ms com allShortestPaths
Quantidade de vínculos obtidos	1	1

A execução no MySQL precisou verificar cada possibilidade no banco para a consulta e retornou um vínculo em quase 20 horas de execução, conforme apresentado na Tabela 2. No Neo4j, antes de utilizar a função de menor caminho, tentou-se executar a Consulta 2 para retornar também todos os relacionamentos, mas após quase 14 horas de execução o processo terminou com estouro de memória. Já utilizando a função de menor caminho, a consulta retornou o mesmo relacionamento encontrado pelo MySQL em pouco mais de 300ms.

Como o objetivo dessa consulta é detectar o relacionamento existente entre as empresas, esse resultado já é suficiente e apresenta um bom indicativo da possibilidade de ganho que um banco de grafos pode oferecer quando utilizado os algoritmos que tiram proveito de sua estrutura.

V. CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho explorou a busca de indícios de fraude em licitações pela detecção de vínculos entre empresas por seus sócios utilizando-se um banco relacional e de grafos. Conforme apresentado, o banco de dados NoSQL de grafos permite aproveitar a estrutura de grafo para aplicações onde o relacionamento entre entidades é tão importante quanto os atributos que as descrevem.

Utilizando-se uma consulta definida pelos auditores da CGU, evidenciou-se a maior simplicidade na busca em profundidade. Verificou-se também um potencial de ganho no

desempenho pelo uso da função *allShortestPath*, permitindo diminuir o tempo de 20 horas para pouco mais de 300 ms.

Como trabalhos futuros, pretende-se expandir a avaliação sobre dados reais para comparar o desempenho; avaliar outras abordagens para o banco relacional, substituindo por tabelas representando vértices e arestas; e um estudo do processo de modelagem de grafos, como o uso de melhores práticas e estudos de caso.

REFERÊNCIAS BIBLIOGRÁFICA

- [1] R. Angles. A Comparison of Current Graph Database Models. In 2012 IEEE 28th International Conference on Data Engineering Workshops (ICDEW), pages 171–177, Apr. 2012.
- [2] S. Batra and C. Tyagi. Comparative analysis of relational and graph databases. International Journal of Soft Computing and Engineering (IJSCE), 2(2), 2012.
- [3] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst., 26(2):4:1–4:26, June 2008.
- [4] M. Chen, S. Mao, and Y. Liu. Big Data: A Survey. Mobile Networks and Applications, 19(2):171–209, Apr. 2014.
- [5] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's Highly Available Key-value Store. In Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles, SOSP '07, pages 205–220, New York, NY, USA, 2007. ACM.
- [6] REF.
- [7] R. Hecht and S. Jablonski. NoSQL evaluation: A use case oriented survey. In 2011 International Conference on Cloud and Service Computing (CSC), pages 336–341, 2011.
- [8] R. C. v. d. Hulst. Introduction to Social Network Analysis (SNA) as an investigative tool. Trends in Organized Crime, 12(2):101–121, June 2009.
- [9] K. Kaur and R. Rani. Modeling and querying data in NoSQL databases. pages 1–7. IEEE, Oct. 2013.
- [10] N. Leavitt. Will NoSQL Databases Live Up to Their Promise? Computer, 43(2):12–14, Feb. 2010.
- [11] J. S. McIlwain. Organized crime: A social network approach. Crime, Law and Social Change, 32(4):301–323, Dec. 1999.
- [12] I. Robinson, J. Webber, and E. Eifrem. Graph Databases. O'Reilly Media, Sebastopol, Calif., 1 edition edition, June 2013.
- [13] S. Srinivasa. Data, Storage and Index Models for Graph Databases. In S. Sakr and E. Pardede, editors, Graph Data Management, pages 47–70. IGI Global, 2011.
- [14] M. Stonebraker. SQL Databases V. NoSQL Databases. Commun. ACM, 53(4):10–11, 2010.
- [15] J. Xu and H. Chen. Criminal Network Analysis and Visualization. Commun. ACM, 48(6):100–107, June 2005.