Pós-Graduação em Ciência da Computação

Paulo Orlando Vieira de Queiroz Sousa

Um Modelo Multidimensional em Padrões de Grafo para Realizar Consultas Analíticas e Topológicas de Grafos Agregados

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2019

Paulo Orlando Vieira de Queiroz Sousa

Um Modelo Multidimensional em Padrões de Grafo para Realizar Consultas Analíticas e Topológicas de Grafos Agregados

Tese apresentada ao Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciências da Computação.

Área de concentração: Banco de Dados.

Orientador: Profª. Ana Carolina Salgado.

Recife

2019

Catalogação na fonte
Bibliotecária Mariana de Souza Alves CRB4-2105

S725m Sousa, Paulo Orlando de Queiroz
Um Modelo Multidimensional em Padrões de Grafo para Realizar Consultas Analíticas e Topológicas de Grafos Agregados/ Paulo Orlando de Queiroz Sousa – 2019.
178 f., fig.; tab.

Orientadora: Ana Carolina Salgado.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação. Recife, 2019.
Inclui referências e apêndices.

1. Banco de Dados. 2. Redes de Informação. 3. OLAP. 4. Análise em Rede. I. Salgado, Ana Carolina (orientadora). II. Título.

025.04 CDD (22. ed.) UFPE - CCEN 2020-13

Paulo Orlando de Queiroz Sousa

“Um Modelo Multidimensional em Padrões de Grafo para Realizar Consultas Analíticas e Topológicas de Grafos Agregados”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 30/08/2019.

_____

Orientador: Profa. Dra. Ana Carolina Brandão Salgado

BANCA EXAMINADORA

_____

Profa.Dra. Valéria Cesário Times
Centro de Informática/UFPE

_____

Prof .Dr. Fernando da Fonseca de Souza
Centro de Informática / UFPE

_____

Prof. Dr. Robson do Nascimento Fidalgo
Centro de Informática / UFPE

_____

Profa. Dra. Maria Luiza Machado Campos
Departamento de Ciência da Computação / UFRJ

_____

Prof. Dr. Carlos Eduardo Santos Pires
Departamento de Sistemas e Computação / UFCG

AGRADECIMENTOS

Esta tese não teria se tornado realidade se eu não recebesse a contribuição de muitas pessoas e também se não abdicasse de alguns momentos da vida. Presto os meus sinceros agradecimentos à Deus, por ter iluminado o meu caminho e ter me dado saúde e perseverança para não desistir e enfrentar as nuances da vida.

Agradeço imensamente a minha professora Carol por ter me acompanhado desde a graduação com a iniciação científica em quase 10 anos de apoio, paciência e críticas. Antes como aluno e hoje como professor aprendi muito com você, Carol, não só pelas lições acadêmicas, mas também pela pessoa cativante que é um exemplo para todos nós.

Gostaria de agradecer às pessoas que sempre estão do meu lado em todas as circunstâncias: minha família, em especial aos meus pais e a minha irmã, que me ajudaram a superar os obstáculos da vida; a minha querida esposa, Aline, que com paciência e companheirismo entendeu minhas abdicações e me ajudou no percurso da tese e aos meus filhos, bênçãos em minha vida.

Por fim, não menos importante, agradeço aos meus amigos e a todos os colegas de trabalho que me ajudaram e apoiaram durante minha trajetória. As instituições de fomento que me auxiliaram no desenvolvimento da pesquisa. A todos muito obrigado pelo apoio contínuo, pelo incentivo incansável e pela compreensão que me ajudou neste trabalho.

RESUMO

Muitos sistemas reais produzem dados em rede ou dados altamente interconectados, os quais podem ser representados em grafo. Os dados das redes de informação formam um componente crítico na infraestrutura da informação moderna, constituindo um grande volume de dados em grafo. A análise das redes de informações abrange diversas áreas, entre elas são destaques as tecnologias OLAP e as análises associadas à estrutura (ou topologia) do grafo. OLAP é uma tecnologia que permite a análise multidimensional e em vários níveis de granularidade, fornecendo visualizações de dados agregados com diferentes perspectivas. A combinação dos algoritmos de análise topológica do grafo com as tecnologias OLAP está em evidência, corroborando com os estudos recentes na área de Grafo BI, ou *Business Intelligence* em grafo. Considerando esse cenário, definimos um modelo multidimensional em grafo para ser usado junto a um SGBDG (Sistema Gerenciador de Bancos de Dados em Grafo). Esse modelo permite analisar padrões em grafo por meio de consultas multidimensionais, combinando os algoritmos de análise em grafo e os operadores OLAP para fornecer uma visualização multidimensional de grafos agregados. Em comparação com as abordagens da literatura, nossa proposta é a primeira a combinar esses recursos de análise nas consultas de um SGBDG. No desenvolvimento deste trabalho, implementamos um *plugin* para o SGBDG Neo4j que permite a produção de consultas multidimensionais analíticas e topológicas em padrões de grafo. Os resultados dessas consultas são representados visualmente, mostrando tanto valores agregados das medidas topológicas e analíticas, quanto a estrutura do grafo agregado. Na experimentação, utilizamos os dados do DBLP para compor diferentes modelagens, a fim de retratar diferentes variações de consulta com grafo agregado.

Palavras-chave: Redes de Informação. OLAP. Análise em Rede. Grafo BI.

ABSTRACT

Many real systems produce networked or highly interconnected data, which can be represented in a graph structure. Information network data is a critical component of modern information infrastructure, constituting a large volume of graph data. The analysis of information networks covers several areas, amongst them are the OLAP technologies and the analysis associated with the structure (or topology) of the graph. OLAP is a technology that enables multi-dimensional and multi-level analysis of granularity, providing aggregate data views with different perspectives. The combination of graph topological analysis algorithms with OLAP technologies is in evidence, corroborating recent studies in the area of Graph BI, or business intelligence in graph. Considering this scenario, we have defined a multidimensional graph model to be used in conjunction with a Graph Database Management System (GDBMS). This model allows one to analyze graph patterns through multidimensional queries, combining graph analysis algorithms and OLAP operators to provide a multidimensional view of aggregate graphs. Compared to literature approaches, our proposal is the first to combine these parsing capabilities into queries of a GDBMS. In the development of this work, we implemented a plugin for the GDBMS Neo4j that allows the production of analytical and topological multidimensional queries in graph patterns. The results of these queries are illustrated, showing both aggregate values of topological and analytical measurements, and the structure of the aggregate graph. In experimentation, we use DBLP data to compose different models in order to portray different query variations with aggregate graph.

Keywords: Information Networks. OLAP. Network Analysis. Graph BI.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

SGBDG	Sistemas de Gerenciamento de Banco de Dados em Grafo
OLAP	*On Line Analytical Processing*
BI	*Business Intelligence*
APOC	*Awesome Procedures on Cypher*
ETL	*Extract Transform Load*
DW	*Data Warehouse*
RDF	*Resource Description Framework*
W3C	*World Wide Web Consortium*
AAMPGrafo	Abordagem de Análise Multidimensional em Padrões de Grafo
Modelo CMPGrafo	Modelo de Consulta Multidimensional em Padrões de Grafo
Modelo MPGrafo	Modelo Multidimensional em Padrões de Grafo
Modelagem MPGrafo	Modelagem Multidimensional em Padrões de Grafo

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo evidenciamos o contexto em que se insere a pesquisa deste trabalho. Para isso, inicialmente descrevemos as motivações para a realização desta pesquisa bem como a problemática investigada. A seguir apresentamos as questões e os objetivos da pesquisa. Por fim, delineamos a organização do documento.

1.1 MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA

Vivemos em um mundo conectado, no qual a maioria dos dados, agentes individuais, grupos ou componentes estão interligados ou interagem uns com os outros, formando redes numerosas, interconectadas e sofisticadas de dados (SUN e HAN, 2012, 2013). Assim, os sistemas do mundo real geralmente consistem em uma grande quantidade de componentes interativos que promovem um crescimento constante de massivas coleções de dados, tais como redes sociais, bioquímicas, ecológicas, de citação, de comunicação, de mobilidade e de transporte (NEWMAN, 2018). Em tais sistemas, os componentes que interagem constituem dados em rede, os quais podem ser chamados de redes de informação (SHI et al., 2015).

A representação da informação em grafo tem o benefício de revelar informações valiosas baseadas no conteúdo e na topologia das redes de informação. Os grafos são estruturas fundamentais e difundidas que fornecem uma abstração intuitiva para a modelagem e análise de dados complexos, heterogêneos e altamente interconectados. O grande poder expressivo dos grafos, juntamente com seu sólido alicerce matemático, encoraja seu uso para modelar estruturas complexas de dados (GHRAB et al., 2018).

Muitas empresas e empreendedores estão interessados em expandir as formas de análise para considerar as características das redes de informação. Nessa perspectiva, a exploração da informação estrutural proveniente das relações entre entidades é um desafio da atualidade que desperta o interesse empresarial (GUMINSKA e ZAWADZKA, 2018). Para a empresa de pesquisa e consultoria Gartner, a análise em grafo é considerada “possivelmente o único diferencial competitivo mais eficaz para as organizações que buscam operações e decisões baseadas em dados após o projeto da captura de dados” (HEUDECKER e FEINBERG, 2014).

No cenário atual, grandes grafos complexos surgiram em vários campos e as análises em grafo estão sendo cada vez mais usadas para resolver problemas complexos. Com isso, as propriedades topológicas do grafo se tornam potenciais repositórios para os sistemas de

tomada de decisão. Esses repositórios fornecem aos sistemas uma nova classe de fatos e medidas de negócios que permitem explorar a estrutura complexa da informação para tomar decisões mais precisas em uma organização orientada a dados (GHRAB et al., 2018).

Os sistemas de *Business Intelligence* (BI) são críticos para a tomada de decisões estratégicas. O Grafo BI, em particular, está emergindo como um campo do BI que integra recursos de análise de rede (GHRAB et al., 2018). Além disso, como a tecnologia OLAP (*On Line Analytical Processing*) já é difundida nos sistemas BI, as empresas estão interessadas em expandir os recursos OLAP nos modelos de dados em grafo (GUMINSKA e ZAWADZKA, 2018).

A necessidade dessa forma de análise, motivou a produção de diversos trabalhos (LOUDCHER et al., 2015; QUEIROZ-SOUSA e SALGADO, 2019), que abrangem análises estruturais (ou topologia) do grafo e tecnologia OLAP. As técnicas de Análise de Rede e Teoria dos Grafos são amplamente difundidas para realizar análises topológicas em grafo, extraindo informações dos relacionamentos que compõem a estrutura do grafo (BRANDES e ERLEBACH, 2005; DIESTEL, 2005). As tecnologias OLAP atuam na análise de dados históricos e de padrões, os quais são considerados promissores na descoberta de conhecimento em grafo de dados (LEE et al., 2016).

Além dessas formas de análise, o *meta-path* é um mecanismo que tem sido adotado nos trabalhos dessa área para melhorar a qualidade da análise de grafos (YIN, Dan et al., 2016). Esse mecanismo utiliza padrões de caminhos (*meta-path*) em grafo para identificar e analisar padrões em grafo. No contexto de Sistemas de Gerenciamento de Banco de Dados em Grafo (SGBDG), o uso de padrões em grafo também é considerado um recurso importante para a análise de grafo. Os SGBDG permitem recuperar dados que correspondam aos padrões especificados nas consultas (ANGLES, 2012).

A integração de algoritmos de análise de rede, de recursos OLAP e de padrões de grafo no SGBDG pode resultar em melhores análises e, consequentemente, decisões mais precisas. A combinação das diferentes técnicas de análise em uma mesma consulta corrobora com o campo de Grafo BI, pois auxilia a tomada de decisão combinando análises topológicas e analíticas em padrões de redes de informação.

Diante desse cenário, consideramos esse campo de pesquisa bastante promissor com grandes possibilidades de crescimento, visto que muitos conceitos estão em aberto e não existe um consenso entre os trabalhos para lidar com esses dados. A análise e a representação visual desse tipo de dados buscam novas formas de extração de

conhecimento que lidem tanto com a informação estrutural, quanto com o conteúdo do grafo de dados.

1.2 QUESTÕES DE PESQUISA E HIPÓTESE DA PESQUISA

Este trabalho tem como objetivo definir uma forma de análise que permita integrar algoritmos de análise de rede, tecnologias OLAP e padrões de grafo às consultas de um SGBDG. A combinação desses diferentes tipos de análise pode melhorar a extração de conhecimento, proporcionando decisões mais precisas em redes de informações. Além disso, a visualização da topologia do grafo e das medidas de análise são fatores importantes para a representação dos resultados de consultas multidimensionais em grafo. Nesta perspectiva, apoiada nas colocações expostas, esta pesquisa cumpre questionar:

Q1 Como integrar algoritmos de análise de rede e tecnologias OLAP a um SGBDG, de modo que permita realizar consultas multidimensionais em padrões de grafo?

Q2 Como representar visualmente os resultados das consultas multidimensionais em padrões de grafo?

Objetivando responder a essas questões, torna-se necessário compreender as principais abordagens que aplicam a tecnologia OLAP na análise de redes de informação. Além disso, conhecer os recursos dos SGBDG, a representação de padrões em grafo e os algoritmos de análise de rede para viabilizar a integração desses em uma abordagem de análise. Assim, definimos uma hipótese para direcionar o desenvolvimento da pesquisa considerando essas questões levantadas.

***Hipótese**: A estruturação do grafo de dados em um modelo de subgrafos orientado a dimensão e a definição de um modelo de consulta possibilitam o SGBDG a realizar consultas que combinem análises topológicas e analíticas em padrões de grafo.*

1.3 DESCRIÇÃO DA PROPOSTA

Para alcançar o objetivo deste trabalho, definimos um modelo multidimensional e um modelo de consulta em grafo para serem usados junto aos SGBDG de modo que possibilitem a realização de consultas analíticas em padrões de grafo. O modelo multidimensional em grafo é responsável por estabelecer uma estrutura orientada ao assunto sobre o foco da análise, o qual é representado por padrões em grafo. Com isso, é estabelecida uma estrutura propícia à realização de consultas topológicas e analíticas.

As consultas dessa abordagem são multidimensionais, atendendo a um modelo que

permite analisar a topologia e as propriedades do grafo (medidas) a partir de diferentes perspectivas (dimensões). Essas consultas são parametrizadas podendo combinar diferentes formas de análise embasadas em algoritmos de análise em grafo, funções de agregação e operadores OLAP na execução da consulta. Os resultados dessas consultas são representados visualmente, mostrando tanto valores agregados das medidas topológicas e analíticas, quanto a estrutura topológica em grafo agregado.

Com a combinação desses recursos, a abordagem poderia analisar em uma rede social, por exemplo, a média salarial das dez pessoas ou grupos de pessoas mais influentes. Esse tipo de consulta utilizaria os padrões do grafo para representar pessoas ou grupos de pessoas no grafo de dados, os algoritmos topológicos para identificar a influência das pessoas ou dos grupos de pessoas na rede social e as funções de agregações para agregar os valores dos salários. Esse capacidade de análise, corrobora com os recentes estudos na área de Grafo BI utilizando os recursos do SGBDG com os algoritmos de análise topológica e analítica em padrões de redes de informação.

1.4 CONTRIBUIÇÕES

Esta tese apresenta as seguintes contribuições:

I. A definição de um Modelo Multidimensional em padrões de grafo, que modela o grafo de dados em uma estrutura multidimensional;
II. A definição de um Modelo de Consulta Multidimensional em padrões de grafo, que permite análises topológicas e analíticas em consultas multidimensionais em grafo;
III. A especificação e implementação de um *plugin* para inserir funcionalidades em um SGBDG que permita tanto realizar consultas multidimensionais em padrões de grafo como processar medidas topológicas e analíticas; e
IV. A implementação de uma representação visual que retrate os resultados das consultas multidimensionais em padrões de grafo, de forma que exiba a agregação topológica do grafo e os valores das medidas de análise.

1.5 ORGANIZAÇÃO DO DOCUMENTO

A organização do restante desse texto é descrita abaixo.

O Capítulo 2 apresenta os conceitos fundamentais que envolvem a tecnologia OLAP, a representação de dados em grafo, as métricas de análise em grafo e os Sistemas

Gerenciadores de Bancos de Dados em Grafos (SGBDG). Além disso, introduzimos o SGBDG Neo4j que foi utilizado na implementação deste trabalho.

O Capítulo 3 mostra uma revisão sistemática no campo de análise em redes de informação, destacando a aplicação da tecnologia OLAP. Com a identificação dos trabalhos, detalhamos as abordagens que se aproximam da pesquisa desenvolvida, descrevendo as suas principais características. Por fim, realizamos uma análise comparativa entre os trabalhos levantados.

O Capítulo 4 apresenta uma visão geral da abordagem proposta e detalha a definição do Modelo Multidimensional em Padrões de Grafo (Modelo MPGrafo) que especifica a forma de modelar os dados em SGBDG. Além disso, são apresentados exemplos de aplicações e uma análise do modelo proposto.

O Capítulo 5 descreve o modelo de Consulta Multidimensional em Padrões de Grafo (CMPGrafo) que especifica os parâmetros de configuração de uma consulta e define as etapas de processamento na realização da consulta junto ao SGBDG.

O Capítulo 6 apresenta uma instância da nossa abordagem sobre o SGBDG Neo4j. Na descrição da implementação é detalhado um *plugin* que incorpora nas consultas do SGBDG recursos de análise topológica e analítica. Além disso, o *plugin* apresenta uma representação na interface gráfica do Neo4j que retrata os resultados das consultas multidimensionais em grafo.

Por fim, o Capítulo 7 apresenta as considerações finais sobre os resultados obtidos, além de apontar futuras pesquisas no desenvolvimento deste campo.

2 FUNDAMENTAÇÃO CONCEITUAL

Este capítulo revisa os fundamentos conceitual necessários para a compreensão das abordagens que analisam o grafo de dados, utilizando as tecnologias OLAP e as métricas de análise em grafo. Nosso objetivo é introduzir os conceitos abordados nos capítulos subsequentes detalhando a tecnologia OLAP, a representação de dados em grafo, as métricas de análise em grafo e os Sistemas Gerenciadores de Bancos de Dados em Grafos (SGBDG).

Com base nessa descrição, organizamos este capítulo da seguinte forma. A Seção 2.1 introduz os conceitos básicos de Sistemas de Data Warehouse. A Seção 2.2 detalhas os conceitos da tecnológica OLAP e do Cubo de dados. A Seção 2.3 define o conceito de redes de informação e aborda modelos de dados em grafo. A Seção 2.4 apresenta as principais métricas de análise em grafo. A Seção 2.5 apresenta as características de um

SGBDG e os seus recursos para manipular um grafo de dados. Finalmente, a Seção 2.6 conclui o capítulo com as considerações finais.

2.1 SISTEMA DE DATA WAREHOUSING

O Sistema de *Data Warehousing* corresponde a um sistema de apoio à decisão que concentra as informações de registros históricos em um repositório, denominado *Data Warehouse*, e permite aos usuários obter respostas de consultas complexas de modo eficiente e preciso (VAISMAN, Alejandro e ZIMÁNYI, 2014). Esses sistemas analíticos, ou Online Analytical Processing (OLAP) são caracterizados por fornecer subsídios para tomadas de decisão a partir de análises sobre as bases de dados históricas com grande volume de dados (INMON, 1993; RASLAN e CALAZANS, 2014). Os dados analisados podem ser acessados sob uma larga possibilidade de visões, de forma rápida, consistente e interativa (VASILAKIS et al., 2004).

2.1.1 Data Warehouse

O *Data Warehouse* corresponde uma coleção de dados orientados a assuntos, integrados, não voláteis e com variação de tempo para apoiar as decisões de gerenciamento (CHAUDHURI et al., 2005). Ele serve para propósitos analíticos e adota o modelo multidimensional, cujos dados são manipulados para a realização de consultas OLAP. Essas consultas são multidimensionais, de modo que permitem analisar dados específicos (medidas) a partir de diferentes perspectivas (dimensões) sobre o *Data Warehouse* (KIMBALL e ROSS, 2013; ROZEVA e ANNA, 2007).

2.1.2 Modelo Multidimensional

A concentração de dados de diferentes origens para a realização de consultas OLAP orientadas a assuntos exige do *Data Warehouse* uma estrutura que viabilize essa forma de análise. Para atender essa exigência, o modelo multidimensional utiliza alguns conceitos (INMON, 2005; VAISMAN, Alejandro e ZIMÁNYI, 2014):

- Fato - representa o foco da análise e normalmente inclui atributos chamados medidas;
- Medidas - geralmente são representados por valores numéricos que permitem uma avaliação quantitativa de vários aspectos de uma organização;

- Dimensões - são usadas para ver as medidas sobre várias perspectivas. Por exemplo, uma dimensão de tempo pode ser usada para analisar as alterações nas vendas durante vários períodos de tempo; e
- Hierarquia - nas dimensões existem geralmente atributos que formam hierarquias, que permitem aos usuários explorar medidas em vários níveis de detalhes. Por exemplo, as hierarquias mês-trimestre-ano na dimensão de tempo ou cidade-estado-país na dimensão de local.

Com base nesses conceitos, o modelo multidimensional é geralmente representado em tabelas relacionais, seguindo a estrutura de um esquema estrela ou esquema floco de neve. Esses esquemas nos bancos de dados relacionais definem uma tabela de fatos (*Fact Table*) e várias tabelas de dimensão (*Dimension Tables*) (VAISMAN, A, 2014). O esquema estrela usa uma tabela exclusiva para cada dimensão, mesmo na presença de hierarquias, o que gera tabelas de dimensões desnormalizadas. Por outro lado, os esquemas de floco de neve usam tabelas normalizadas para dimensões e suas hierarquias.
Com a composição do *Data Warehouse*, uma ferramenta OLAP é utilizada para permitir consultas analíticas e fornecer visualizações multidimensionais do *Data Warehouse*.

2.2 ONLINE ANALYTICAL PROCESSING

Uma ferramenta OLAP é constituída por um conjunto de técnicas especialmente projetadas para auxiliar o processo de consultas, análises e cálculos dos dados, proporcionando condições de análise de grande volume de dados.

2.2.1 Operações OLAP

Uma das funções que devem estar presentes na ferramentas OLAP é a capacidade de efetuar algumas operações, tais como (JIAWEI et al., 2012):

- Roll-up - executa a agregação de medidas ao subir o nível hierárquico de uma dimensão, obtendo valores de agregação com maior granularidade (nível maior de generalização);
- Drill-down - executa a agregação de medidas ao descer o nível hierárquico de uma dimensão, obtendo valores de agregação com menor granularidade (nível maior de detalhe);
- Slice - restringe os dados a serem analisados utilizando uma dimensão para fixar um valor na visualização da consulta; e

- Dice - restringe os dados a serem analisados utilizando duas ou mais dimensões para selecionar um faixa de valores na consulta.

2.2.2 Ferramentas OLAP

As ferramentas OLAP são aplicações que permeiam diversas camadas da tecnologia, indo do armazenamento até as camadas de linguagens. Essas ferramentas são classificadas de acordo com a arquitetura e os objetivos (PAMPA QUISPE, 2003; VAISMAN, Alejandro e ZIMÁNYI, 2014), destacando as seguintes classificações:

- ROLAP (*Relational OLAP)* - os dados são armazenados em banco de dados relacionais (RDB – *Relational Databases*). Não possui dados pré-processados e utiliza um servidor multidimensional para transformar as consultas em resultados OLAP. Além disso, permite o armazenamento de dados novos sem precisar de pré-processamento;
- MOLAP (*Multidimensional OLAP)* - permite a execução de análises sofisticadas, usando bancos de dados multidimensionais (Multidimensional Databases - MDB). Os dados são pré-processados e modelados em uma estrutura multidimensional; e
- HOLAP (*Hybrid OLAP)* - é um hibrido de ROLAP e MOLAP, podendo realizar consultas OLAP tanto em RDB quanto em MDB, juntando as vantagens das duas abordagens.

A escolha da ferramenta OLAP repercute nas características do sistema OLAP, visto que o ROLAP atende sistemas que requerem dados atuais podendo recuperá-los diretamente das fontes de dados, enquanto que o MOLAP utiliza dados pré-processados do cubo para obter resultados de consulta em menor tempo, de modo que o repositório utilizado não oferece suporte as mudanças recorrentes nos dados.

2.2.3 Implementação do Cubo de Dados

Um cubo de dados permite que os dados sejam modelados e visualizados em função de várias dimensões, sendo assim considerado um cubo n-dimensional (JIAWEI et al., 2012). Além disso, o cubo de dados tem o intuito de reduzir o tempo de resposta de consultas complexas, as quais cruzam vários conjuntos de dimensões durante a agregação de dados.

A implementação de um cubo de dados envolve o pré-processamento de todos os possíveis resultados de agrupamento de dados em razão das possíveis combinações de

dimensões. Em SQL, estas agregações são realizadas por consultas do tipo *group_by*. Na composição do cubo, as agregações de dados sobre as combinações de dimensões são chamadas de cuboid. Na construção de um cubo, podemos gerar um cuboid para cada subconjunto possível das dimensões consideradas. Esse resultado formaria uma estrutura *lattice* de cuboids, a qual organiza os dados em níveis diferentes de agregações. Por fim, essa estrutura *lattice* de cuboids corresponde a um cubo de dados (GORUNESCU, 2011).

A Figura 1 apresenta um exemplo de estrutura *lattice* de um cubo de dados que contém três dimensões {cidade, item e ano} e um medida de vendas. Esse cubo possui 2^3 = 8 cuboids no total, formando um cuboid para cada subconjunto das dimensões. O cuboid base é o menos generalizado (mais específico) dos cuboids, enquanto que o cuboid 0-D (apex) é o mais generalizado (menos específico) dos cuboids, e muitas vezes denotado como ALL. Se começarmos no cuboid apex e navegarmos para baixo na estrutura lattice, isso equivale à operação *drill-down* dentro do cubo de dados. Consequentemente, navegar para cima corresponde à operação *roll-up* (GORUNESCU, 2011).

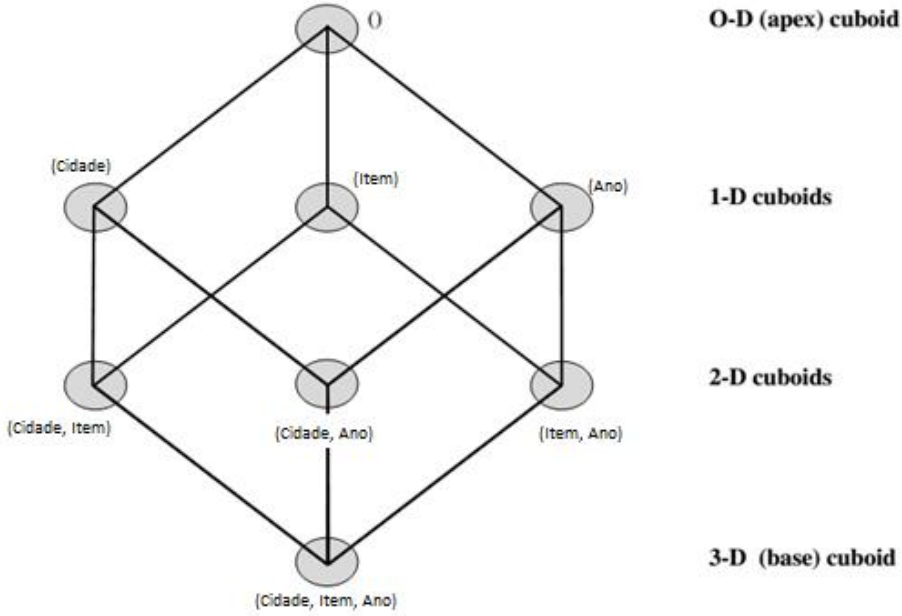


Figura 1 - Exemplo de uma estrutura Lattice de cuboids (GORUNESCU, 2011)

Com base na produção e no armazenamento do cubo de dados, diferentes formas de materializar o cubo foram definidas.

2.2.4 Materialização do Cubo de Dados

Na literatura, encontramos três formas de realizar materialização do cubo de dados:

- ***No materialization*** - não precisa armazenar todos os agrupamentos, sendo necessário processar os cuboids durante a consulta. Essa estratégia resulta em consultas lentas com dados recentes, pois os dados são recuperados em tempo real;
- ***Full materialization*** - Essa forma de implementação produz um cubo completo na estrutura lattice com todos os cuboids computados. Esta escolha requer uma grande quantidade de espaço para armazenar todos os cuboids pré-processados; e
- ***Partial materialization*** - processa seletivamente um subconjunto dos possíveis cuboids, considerando os critérios do usuário na seleção dos cuboids a serem pré-computados. Essa abordagem representa um equilíbrio entre espaço de armazenamento e tempo de resposta.

Considerando o grande custo computacional e o espaço de armazenamento, várias ferramentas OLAP adotaram abordagens heurísticas para a materialização parcial do cubo. Uma dessas abordagens é o cubo de *iceberg*, que consiste em processar um cubo de dados e armazenar apenas as células, as quais possuem um valor de agregação que atendem aos critérios do usuário (GORUNESCU, 2011).

2.3 REPRESENTAÇÃO DE DADOS EM GRAFO

Nessa seção, apresentamos os conceitos básicos para representar os dados das redes de informações utilizando modelos de dados em grafo. Nos conceitos apresentados a seguir são considerados os seguintes conjuntos:

- **O** um conjunto de objetos;
- **V** $\subset$ **O** é um conjunto finito de objetos, chamado vértices;
- **A** $\subset$ **O** é um conjunto finito de objetos, chamado arestas;
- **Rv** um conjunto finito de rótulos do tipo vértice;
- **Ra** um conjunto finito de rótulos do tipo aresta;
- **(Rv** $\cup$ **Ra)** $\subseteq$ **R** um conjunto finito com todos os rótulos;
- **K** um conjunto de propriedades chave e
- **N** um conjunto de valores.

Esses conjuntos são utilizados na definição dos conceitos e na representação estrutural do grafo.

2.3.1 Conceitos Básicos de Grafo

Formalmente, um grafo simples é constituído por um par de conjuntos (V, A), onde V é um conjunto finito de vértices, A é um conjunto finito de arestas e $V \cap A = \emptyset$. Cada aresta do conjunto A é constituída por pares de vértices não ordenados, de modo que uma aresta com os vértices u e v pode ser representada tanto por {u,v} quanto {v,u}. (BONDY e MURTY, 1976; RUOHONEN, 2013).

Com base nessa definição básica do grafo, é possível manipular e definir outros modelos de grafo utilizando alguns conceitos específicos que detalhamos a seguir:

- **Grafo Direcionado** - define uma direcionamento nas arestas formando um grafo direcionado (dígrafo). As arestas são compostas por pares ordenados de vértices, onde um aresta que liga os vértices u e v pode ser representada de duas formas diferentes tanto {u,v} como no sentido oposto {v,u} (RUOHONEN, 2013);
- **Grafo Rotulado** - apresenta uma marcação (rótulo) nos elementos do grafo G = (V, A) representando um mapeamento $\alpha: V \rightarrow Rv$, onde Rv é chamado de conjunto de rótulos dos vértices. Da mesma forma, uma marcação das arestas é um mapeamento $\beta: A \rightarrow Ra$, onde Ra é o conjunto de rótulos das arestas (RUOHONEN, 2013);
- **Multigrafo** - permite múltiplas arestas entre os pares de vértices;
- **Grafo de propriedade** - permite enriquecer os vértices e as arestas do grafo com dados no formato de pares de chave-valor, denominado de propriedade. Alguns modelos em grafo podem aplicar propriedades em vértices ou arestas apenas;
- **Caminho (Path)** - é um caminho que representa um trajeto de modo que forma um caminho sem arestas e sem vértices repetidos; e
- **Subgrafo** - um subgrafo de um grafo G é qualquer grafo H, tal que $V(H) \subseteq V(G)$ e $A(H) \subseteq A(G)$, de modo que se $H \subseteq G$ e $H \neq G$, então $H \subset G$ e H é considerado um subgrafo.

Considerando os conceitos básicos do grafo, retratamos a seguir o conceito de redes de informações e a representação de dados em grafo.

2.3.2 Redes de informação

Muitos sistemas reais geralmente possuem um grande número de componentes interativos de vários tipos, como atividades sociais humanas, sistemas de comunicação e computação, e redes biológicas. Em tais sistemas, os componentes que interagem

constituem redes interconectadas, que podem ser chamadas de redes de informação sem perda de generalidade (SHI e YU, 2017). Uma rede de informação representa uma abstração do mundo real, retratando informações que podem ser representadas por objetos e interações entre esses objetos.

2.3.2.1 Definição Formal

Formalmente, uma rede de informação é definida por um grafo direcionado G = (V, A), contendo uma função para mapear o tipo de objeto (rótulo do vértice) ϕ: V → Rv e uma função para mapear o tipo de relacionamento (rótulo da aresta) ψ: A → Ra. Cada objeto v ∈ V pertence a um tipo particular de objeto identificado pelos rótulos do vértice Rv: ϕ (v) ∈ Rv, da mesma forma cada relacionamento a ∈ A pertence a um tipo particular de relacionamento contendo rótulos de arestas específicos Ra: ψ (e) ∈ Ra. Se dois relacionamentos pertencerem ao mesmo tipo de relacionamento, então os dois compartilham os mesmos tipos de objeto inicial e final no relacionamento (SHI e YU, 2017; SUN e HAN, 2012). Com essas definições, os autores SHI e YU, (2017) diferenciam tipos de objetos e tipos de relacionamentos na rede de informação, permitindo caracterizar a rede de informação em heterogênea, caso os tipos de objetos | Rv | > 1 ou os tipos de relacionamento | Ra | > 1; e em homogênea, caso contrário (SHI e YU, 2017; SUN e HAN, 2012).

2.3.2.2 Esquema de Rede de informação

Para compreender melhor os tipos de objeto e os tipos de relacionamentos em uma rede de informação heterogênea, é necessário uma descrição a nível de esquema da rede. Portanto, foi proposto um conceito de esquema de rede para descrever a meta estrutura de uma rede de informação (SHI e YU, 2017; SUN e HAN, 2012).

O esquema de rede, denotado como EG = (Rv, Ra), é um metamodelo para uma rede de informação G = (V, A), mapeando os tipos de objeto ϕ: V → Rv e os tipos de relacionamento ψ: A → Ra. Esse esquema forma um grafo direcionado com os tipos de objetos Rv e os tipos de relacionamento Ra.

Considerando as características das redes de informação, abordamos dois modelos de dados em grafo para lidar com a interconectividade dos dados e permitir o uso dos recursos OLAP.

2.3.3 Modelo de Grafo de Propriedade

O Modelo de Grafo de Propriedade (MGP) representa os dados como um multigrafo de propriedade direcionado (BONIFATI et al., 2018). Os vértices e arestas são objetos com um conjunto de rótulos e um conjunto de pares de chave-valor, as chamadas propriedades. Para uma definição formal, apresentamos a seguinte estrutura MGP = (V, A, §n, §h, §u) (BONIFATI et al., 2018), onde

- V ⊆ O é um conjunto finito de objetos, chamado vértices;
- A ⊆ O é um conjunto finito de objetos, chamado arestas;
- §n : A ⟶ V x V é uma função que atribui a cada aresta um par ordenado de vértices ;
- §h : (V U A) ⟶ P(R) é uma função que atribui a cada vértice ou aresta um conjunto finito de rótulos. (a função P(R) representa os conjuntos de rótulos);
- §u : (V U A) x K ⟶ N é uma função parcial que atribui valores N para as propriedades chaves K dos objetos (V U A), tal que os conjuntos de objetos V e A são disjuntos (isto é, V ∩ A = ∅).

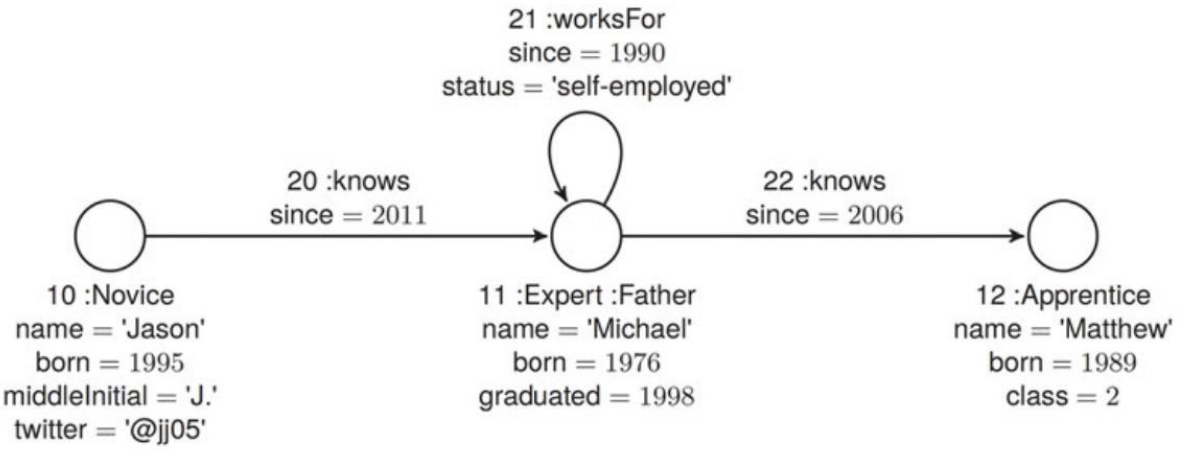


Figura 2 - Exemplo de Grafo de propriedade (BONIFATI et al., 2018)

A Figura 2 mostra um exemplo simples de um grafo de propriedades. O grafo tem os três vértices (10, 11 e 12) e três arestas direcionadas (20, 21 e 22) conectando-os. Os rótulos são informações descritivas de classe que marcam os objetos representando tipos de entidade do mundo real. Os rótulos são prefixados de dois pontos sobre os elementos do grafo, como mostra o vértice 11 contendo dois rótulos :Expert e :Father. As propriedades fornecem os dados reais que um objeto representa. A chave da propriedade especifica o significado do valor da propriedade. Por exemplo, a aresta 22 tem uma propriedade com a chave “since” e o valor 2006. Como os rótulos são puramente descritivos, eles não implicam em uma informação de propriedade e os objetos podem instanciar qualquer propriedade

independente dos seus rótulos.

2.3.4 Modelo de Grafo de Propriedade com Hipervértices

O Modelo de Grafo de Propriedade com Hipervértices (MGPH) é uma extensão do MGP, na qual um hipervértice retrata um subgrafo que tanto pode ser vazio como pode conter um conjunto de vértices interconectados (BONIFATI et al., 2018). Os hipervértices ao denotar subgrafos vazios representam vértices tradicionais, enquanto que os hipervértices denotando subgrafos não vazios retratam objetos de ordem superior. Como os hipervértices podem ser relacionados por meio de arestas, o MGPH permite relacionar um subgrafo a outros subgrafos, considerando até mesmo os hipervértices dentro do próprio subgrafo.

Formalmente, o modelo de grafo de propriedades em hipervértice segue a seguinte estrutura MGPH = (V, A, §n, §y, §h, §u), onde

- $V \subseteq O$ é um conjunto finito de objetos, chamados hipervértices;
- $A \subseteq O$ é um finito conjunto de objetos, chamados arestas;
- §n : $A \longrightarrow V \times V$ é uma função que atribui a cada aresta um par ordenado de vértices;
- §y : $V \longrightarrow P(V) \times P(A)$ é uma função total que atribui a cada hipervértice um conjunto de hipervértices e um conjunto de arestas, de tal modo que:
  - os conjuntos de objetos V e A são disjuntos, ou seja, $V \cap A = \emptyset$;
  - os hipervértices são grafos bem formados no sentido de que todas as arestas do hipervértice v são adjacentes aos hipervértices contidos em v, ou seja, para todo $e \in trg(y(v))$ é o caso que $n(e) \in src(y(v)) \times src(y(v))$;
  - o conjunto de valores de domínio que v é definido é finito
- §h : $(V \cup A) \longrightarrow P(T)$ é uma função que atribui a cada objeto um conjunto finito de rótulos.
- §u : $(V \cup A) \times K \longrightarrow N$ é uma função parcial que atribui valores para as propriedades dos objetos, tal que os conjuntos de objetos V e A são disjuntos (isto é, $V \cap A = \emptyset$).

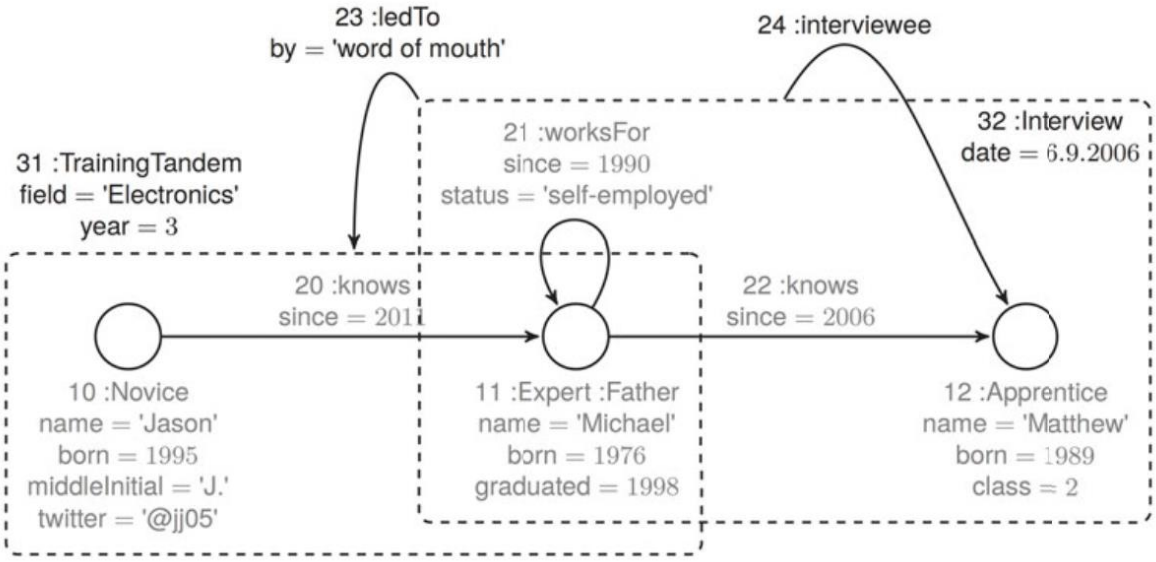


Figura 3 - Exemplo de Grafo de Propriedade com Hipervértices (BONIFATI et al., 2018)

A Figura 3 mostra um grafo de propriedades com hipervértices. O hipervértice 10 :Novice indica um subgrafo vazio. Já o hipervértice 32 :Interview denota um subgrafo não vazio, constituído pelos hipervértices 11 e 12 e pelas arestas 21 e 22. Esse hipervértice 32 tem duas arestas de saída 23 e 24, que o relaciona ao hipervértice 31, com subgrafo não vazio, e ao hipervértice 12, contido no próprio hipervértice 32, respectivamente. Dessa forma, os hipervértices podem se sobrepor e sua união também pode ser um subconjunto do grafo completo.

2.4 ANÁLISE EM GRAFO

Nessa seção, introduzimos as principais métricas da teoria dos grafos para a realização de análise em grafo. A Teoria dos Grafos é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto, constituindo o modelo matemático (grafo) para estudar as relações entre os objetos (BONDY e MURTY, 2010). Análise de Redes é o estudo de representações de rede, a qual utiliza formas específicas para analisar diferentes estruturas, como internet, sistema de transporte público e sistema elétrico (BRANDES e ERLEBACH, 2005).

No âmbito da Teoria dos Grafos e Análise de Redes, existem várias medidas de centralidade para os vértices de um grafo. Uma medida de centralidade determina a importância relativa de um vértice no grafo. Por exemplo, avaliar a importância de uma pessoa dentro de uma rede social ou definir a utilidade de uma estrada dentro de uma rede urbana. Entre as medidas de centralidade existem quatro que são amplamente utilizadas

na análise de rede: *degree centrality*, *betweenness centrality*, *closeness centrality* e *eigenvector centrality* (BORGATTI e EVERETT, 2006; FREEMAN, 1978; NEWMAN, 2010; OPSAHL et al., 2010), detalhamos a seguir essas medidas.

2.4.1 Degree Centrality

Degree Centrality é definida em razão do número de ligações incidentes sobre um vértice, ou seja, o número de ligações que um vértice contém. Em caso de uma rede direcionada é possível definir duas medidas separadas para representar a *Degree Centrality*: *Indegree*, que conta o número de ligações direcionadas ao vértice, e *Outdegree*, que conta o número de ligações direcionadas de um vértice aos outros. A representação da *Degree Centrality* $C_D(v)$ é definida a seguir:

$$C_D(v) = \frac{\deg(v)}{n-1}$$

onde $\deg(v)$ igual ao número de relações para um vértice v e n é o número de vértices contido na rede.

2.4.2 Betweenness Centrality

Betweenness centrality é uma medida de centralidade definida pela ocorrência de um vértice nos menores caminhos do grafo formado pelos outros vértices. Esta medida valoriza os vértices que compõem os menores caminhos entre os vértices. Para um grafo com n vértices, o cálculo do *betweenness* precisa:

1. Determinar para cada par de vértices (α, μ) os menores caminhos entre eles;
2. Determinar para cada par de vértices (α, μ) a fração de menores caminhos que passam sobre o vértice v em questão;
3. Somar todas as frações dos pares de vértices (α, μ).

A medida é representada pela seguinte fórmula:

$$C_B(v) = \sum_{\alpha \neq v \neq \mu \in V} \frac{\sigma_{\alpha\mu}(v)}{\sigma_{\alpha\mu}}$$

onde $\sigma_{\alpha\mu}$ é a quantidade de menores caminhos de α para μ e $\sigma_{\alpha\mu}(v)$ é o número de menores caminhos de α para μ que passam pelo vértice v.

2.4.3 Closeness Centrality

Closeness centrality utiliza a proximidade como um conceito básico do espaço

topológico, que intuitivamente define se dois conjuntos estão arbitrariamente próximos um do outro. Na Teoria dos Grafos, a medida *Closeness* é definida para avaliar a aproximação de um vértice dos outros vértices do grafo. Na análise de rede, essa medida é definida em função da média da distância geodésica (caminho mais curto) entre um vértice v e todos os outros vértices alcançáveis a partir de v:

$$\sum_{\mathrm{t}\in\mathrm{V}\setminus\mathrm{v}} \frac{\mathrm{d_G}(v,t)}{\mathrm{n}-1}$$

onde $\mathrm{n} \geq 2$ e $\mathrm{d_G}(v,t)$ é a distância geodésica dos vértices t alcançáveis por v. Essa fórmula pode ser considerada como uma média de quanto tempo leva para espalhar a informação de um vértice para os outros vértices alcançáveis. A medida *Closeness centrality* $\mathrm{C_C}(v)$ de um vértice v é o inverso da soma da distância geodésica (caminho mais curto) para todos os outros vértices V:

$$C_C(v) = \frac{1}{\sum_{t\in V\setminus v} d_G(v,t)}$$

2.4.4 Eigenvector Centrality

Eigenvector centrality é uma medida que mensura a importância de um vértice na rede. Ela atribui uma pontuação relativa a todos os vértices da rede, considerando a importância das conexões indiretas em todo o comprimento da rede. A medida *eigenvector* utiliza uma matriz quadrada de ordem N (número total de vértices) chamada matriz de adjacência $A_{i,j}$ que registra $A_{i,j} = 1$ para os vértices adjacentes e $A_{i,j} = 0$ caso contrário. A medida de um vértice x_i é proporcional à soma de todos os vértices que estão conectados ao vértice x_i.

$$x_i = \frac{1}{\lambda}\sum_{j=1}^{N} A_{i,j}x_j$$

onde N é o total de vértices e λ é uma constante. Com a definição do vetor de centralidades $x = (x1, x2, \dots)$, podemos reescrever esta equação na forma de matriz, como se segue:

$$\lambda\, x = A.x$$

Dessa forma, x é um autovetor da matriz de adjacência A com um autovalor λ que geralmente corresponde ao maior autovalor. A medida *Eigenvector Centrality* atribui a cada vértice uma centralidade que depende tanto do número de conexões quanto da qualidade das conexões. Com isso, a medida *eigenvector* acaba sendo relevante em várias situações. Por exemplo, uma variante da *Eigenvector Centrality* é empregada no algoritmo *PageRank*

(PAGE et al., 1999) que é usado na classificação de páginas web.

2.5 BANCO DE DADOS EM GRAFO

Os Sistemas Gerenciadores de Bancos de Dados em Grafos (SGBDG) são exemplos de Sistemas NoSQL, os quais modelam e manipulam os dados por meio de vértices e arestas. Esse modelo permite a representação de contextos complexos com grande interconectividade nos dados, destacando os relacionamentos entre as entidades na base de dados (POKORNÝ, 2016).

Na utilização de dados em grafo, existem duas abordagens principais amplamente utilizadas. A primeira consiste em usar nativamente o modelo de grafo nos mecanismos de banco de dados. A segunda utiliza modelos alternativos, principalmente o modelo relacional, para representar um grafo de dados. Essas abordagens são discutidas a seguir (GHRAB et al., 2018):

2.5.1 Abordagem com Grafo Nativo

Nos últimos anos, o desenvolvimento de SGBDG nativo passou a ser uma tendência. A maioria dos SGBDG nativo implementam o modelo de grafo de propriedade ou uma variação dele (GHRAB et al., 2018). Essa abordagem faz com que o modelo de grafo seja implementado tanto no armazenamento físico dos dados, quanto no processamento de consultas. Com isso, aumenta a compatibilidade da consulta com os dados armazenados, tornando as consultas mais intuitivas. Do ponto de vista de desempenho, os SGBDG nativos são otimizados para travessias em grafo. O custo de atravessar uma aresta é constante e o custo de navegação no grafo é menor do que as junções entre as tabelas do modelo relacional.

Os SGBDG nativos, devido ao armazenamento de dados em grafo, permitem tipos específicos de consultas em grafo (ANGLES, 2012), que enumeramos a seguir:

- Consultas adjacentes - o princípio básico é a adjacência vértice / aresta. Dois vértices são adjacentes (ou vizinhos) quando possuem uma aresta entre eles. Da mesma forma, duas arestas são adjacentes quando compartilham o mesmo vértice. Dessa maneira, podemos realizar consultas para validar se dois vértices ou duas arestas são adjacentes, assim como, listar todos os vizinhos de um vértice;
- Consultas de Alcance - este tipo de consultas é caracterizado por problemas de caminho ou de travessia. O problema de alcançabilidade consiste em testar se dois

vértices são conectados por um caminho. Nesse contexto, podemos considerar dois tipos de caminhos: caminhos de comprimento fixo, que contêm um número fixo de vértices e arestas; e caminhos simples regulares, que determinam algumas restrições nos vértices e arestas (por exemplo, expressões regulares);

- Consultas de correspondência de padrões - este tipo de consulta consiste em localizar todos os subgrafos de um grafo de dados que são isomórficos a um dado padrão de grafo; e
- Consultas de sumarização - esse tipo de consulta não está relacionada à estrutura topológica do grafo. Em vez disso, são consultas baseadas em funções especiais que permitem resumir ou computar os valores resultantes da consulta, retornando um único valor. As funções agregadas (por exemplo, avg, count, max) são exemplos desse tipo de consulta.

Com base nessas formas de consulta, percebemos que os SGBDG não-nativos não conseguem explorar alguns desses algoritmos de consulta, pois o modelo relacional foi projetado para executar varreduras de tabela em vez de travessias, não sendo adequado para lidar com a estrutura topológica do grafo.

2.5.2 Abordagem com Grafo Não-Nativo

Essa abordagem utiliza os recursos do modelo relacional para manipular o grafo de conhecimento, o qual é representado em Resource Description Framework (RDF) (CYGANIAK et al., 2014), uma linguagem recomendada pela World Wide Web Consortium (W3C) para descrever e vincular recursos (GHRAB et al., 2018). Nessa abordagem, os dados do RDF são manipulados em tabelas do modelo relacional utilizando triplas <assunto, predicado, objeto> para registrar as informações.

A cobertura do modelo relacional no mercado beneficiou a disseminação do RDF, possibilitando uma integração suave com uma ampla variedade de plataformas relacionais. No entanto, as implementações de grafo no modelo relacional são deficitárias no cumprimento de requisitos para: (i) a modelagem intuitiva dos dados; (ii) as consultas de reconhecimento topológico do grafo (tais como, recuperação e comparação de caminho, e correspondência de padrões em grafo); e (iii) o desempenho otimizado na travessia do grafo.

Nessa abordagem, o mapeamento da estrutura do grafo para as tabelas do modelo relacional causa problemas de incompatibilidade nos níveis de modelagem e consulta. A

travessia de arestas é simulada usando operações caras de junção, o que promove uma carga de trabalho pesada, especialmente para tabelas altamente interconectadas. Além disso, o SQL não é adequado para realizar consultas sobre a topologia do grafo, tendo limitações para realizar correspondência de padrões, recuperação de caminhos e identificação de dados vizinhos.

2.5.3 SGBDG Neo4j

Neo4j é um SGBDG nativo desenvolvido em Java pela Neotechnology para ser integrado nas aplicações ou ser acessado como cliente / servidor via API REST. A manipulação do grafo no Java Neo4j é muito natural, com o uso de sua API é possível modelar o grafo de dados e adicionar conjuntos de propriedade nos vértices e arestas do grafo (MPINDA et al., 2015).

Esse SGBDG tem como característica o suporte a transações (Atomicidade, Consistência, Isolamento e Durabilidade - ACID) (PANZARINO, 2014), alta disponibilidade e alta velocidade em consultas. Utiliza de forma nativa o modelo grafo de propriedades no armazenamento e processamento dos dados. Além disso, o Neo4j possui versões *open-source* e comerciais, oferecendo dois tipos de arquitetura: centralizada e distribuída com suporte à replicação (MILLER, 2013).

O Neo4j possui uma linguagem de consulta nativa denominada Cypher, que foi projetada para facilitar o uso dos desenvolvedores. Embora atualmente específica do Neo4j, essa linguagem permite representar os grafos de maneira intuitiva usando diagramas na descrição dos dados em grafo (CELKO, 2014). O Cypher fornece uma sintaxe declarativa que permite consultas de correspondência de padrões. Esta linguagem relativamente simples, mas poderosa, permite facilmente expressar consultas muito complicadas em bancos de dados (MPINDA et al., 2015).

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, apresentamos uma visão geral sobre a tecnologia OLAP, a representação e análise de dados em grafo, e os SGBDG. Esses conceitos compõem a base deste trabalho. A tecnologia OLAP introduz os principais conceitos para modelar e consultar dados multidimensionais, descrevendo a modelagem multidimensional e os operadores OLAP para a composição das consultas. Introduzimos o conceito de redes de

informação descrevendo as características dos grafos, detalhando dois modelos de grafo para estruturar os dados em grafo. Apresentamos as métricas de análise da teoria do grafo e da análise de rede. Descrevemos as características de um SGBDG, detalhando os principais recursos de um SGBDG nativo e introduzimos o SGBDG Neo4j. Com isso, conseguimos abordar os principais conceitos para auxiliar o entendimento deste trabalho. No capítulo seguinte, apresentamos o estado da arte detalhando os principais trabalhos relacionados da área.

3 ESTADO DA ARTE

Neste capítulo apresentamos uma revisão sistemática realizada para encontrar os principais trabalhos da área e, consequentemente, as melhores soluções para a realização de análises em grafo de dados. Detalhamos os trabalhos que mais se aproximam da pesquisa desenvolvida nesta tese, mostrando suas principais características. Por fim, apresentamos uma análise comparativa entre os trabalhos para discutir as suas principais contribuições.

Este capítulo cobre o seguinte conteúdo. A Seção 3.1 descreve a realização da revisão sistemática. A Seção 3.2 apresenta os principais trabalhos relacionados ao desenvolvimento da tese. A Seção 3.3 mostra uma análise comparativa discutindo as principais características que norteiam o campo da pesquisa. Finalmente, a Seção 3.4 conclui o capítulo evidenciando as considerações finais.

3.1 REVISÃO SISTEMÁTICA

Com base no trabalho de KITCHENHAM e CHARTERS (2007), realizamos uma revisão sistemática para consolidar as evidências existentes sobre a aplicação das tecnologias OLAP em grafo de dados e identificar as potenciais características dessas aplicações. Na realização dessa revisão, definimos um protocolo para coleta de trabalhos na literatura e uma metodologia para identificar os trabalhos relevantes da área. O protocolo consistiu em buscar trabalhos por palavras-chave nas bibliotecas digitais e nas ferramentas de busca específicas à área da computação. Nessas buscas foram utilizadas como base as seguintes palavras-chave: “Graph; Graph-based; Information Network; OLAP; Cube”. Os repositórios utilizados e o número de artigos recuperados estão descritos na Quadro 1.

Os resultados alcançados nas buscas por palavra-chave corresponderam a 2383 artigos que foram submetidos a uma metodologia para identificar os trabalhos mais relevantes. Nessa metodologia utilizamos critérios de inclusão e exclusão para identificar os trabalhos relevantes. Os critérios de inclusão consiste de (i) Estudos primários publicados em revistas ou conferências, que lide com base de dados em grafo e que realize análises com agregação de dados; (ii) Estudos teóricos com o objetivo de apresentar conceitos para o entendimento da área; (iii) Estudos experimentais relacionados à área; e (iv) Considerar apenas trabalhos escritos em língua inglesa. Os critérios de exclusão consiste de (i) Estudos que não estejam claramente relacionados a grafo de dados e agregação de dados; (ii) Artigos duplicados, ou seja, aqueles encontrados em mais de uma

fonte da busca automática e/ou manual; (iii) Estudos não disponíveis para download nos engenhos de busca definidos; e (iv) Estudos secundários, ou seja, que dependam de estudos primários.

A execução da metodologia consiste de duas etapas: uma de seleção, utilizando e outra de extração. Na etapa de seleção realizamos leituras rápidas nos títulos, palavras-chave e *resumos* e selecionamos de 187 artigos com base nos critérios estabelecidos.

Quadro 1 - Fontes e artigos da revisão sistemática

Fontes de busca	Número de Artigos
ACMDigital Library[1]	465
DBLP[2]	195
Engineering Village[3]	466
Google Scholar[4]	107
IEEEXplore Digital Library [5]	83
Elsevier ScienceDirect[6]	406
Springer Link[7]	461
ISI Web of Knowledge[8]	89
Elsevier Scopus[9]	113
Total	**2383**

Na etapa de extração foram realizadas leituras nos títulos, *resumos*, introdução e conclusões para extrair os artigos que não atendiam aos critérios de exclusão estabelecidos. No final do processo, 25 artigos científicos de estudo primário foram identificados na revisão sistemática para formar a base de estudo desta tese e compor um artigo survey (QUEIROZ-SOUSA e SALGADO, 2019). Com esses artigos, e outros que encontramos posteriormente em pesquisas avulsas, selecionamos os artigos mais próximos da pesquisa desenvolvida na tese para detalhá-los na próxima seção.

3.2 TRABALHOS RELACIONADOS

Durante a revisão sistemática observamos várias nomenclaturas para especificar

[1] http://dl.acm.org
[2] http://dblp.uni-trier.de/
[3] http://www.engineeringvillage2.org
[4] https://scholar.google.com
[5] http://ieeexplore.ieee.org/
[6] http://www.sciencedirect.com
[7] http://link.springer.com/
[8] http://www.webofknowledge.com/
[9] http://www.scopus.com

esse tipo de abordagem, que utiliza a tecnologia OLAP para analisar os dados em grafo. Nesse estudo percebemos o quão abertas estão as definições da área. Os trabalhos encontrados apresentam diferentes métodos para analisar, visualizar e selecionar os dados em grafo. Apesar das diferentes abordagens, os trabalhos possuem o mesmo objetivo, promover uma análise que permita agregar os dados de um grafo. Em razão desse objetivo, selecionamos os trabalhos que estão mais relacionados com algumas das principais características desta tese, a qual consiste em: produzir grafo agregado, analisar padrões em grafo e realizar consultas OLAP. A seguir, detalhamos os trabalhos que cobrem algumas dessas características.

3.2.1 Graph OLAP

Nos trabalhos de CHEN et al. (2008, 2009), os autores propuseram um *framework*, conceitual chamado Graph OLAP, o qual permite aplicar OLAP em qualquer parte do grafo de dados. Esse *framework* lida com dados interconectados, de modo que permite tanto a análise de informações, a qual é constituída por valores dos atributos, quanto a análise topológica do grafo de dados, a qual processa a estrutura topológica do grafo. Além disso, esse *framework* permite generalizar / especializar dinamicamente a estrutura do grafo de dados, oferecendo visões multidimensional e múltinível sobre o grafo de dados. Para explanar essas características de análise, os autores mostraram na Figura 4 uma rede de colaboração entre autores.

Nesse trabalho, os autores apresentaram definições básicas de dimensão e de medida, que compuseram a base conceitual para a aplicação do OLAP e do cubo de dados em grafo. As dimensões são usadas para particionar os dados em diferentes células que constituem a estrutura *lattice* de cuboids. Já as medidas são processadas para agregar os dados do grafo, no intuito de fornecer uma visão resumida dos mesmos.

As dimensões são representadas pelos rótulos e propriedades de diferentes instâncias de grafo. Talvez, devido a esta decisão de modelagem, os autores dividiram a forma de aplicar dimensões OLAP em: *informational OLAP* (I-OLAP), que utiliza as informações dos atributos para agregar os vértices de mesmo rótulo, e *topological OLAP* (T-OLAP), que usa a estrutura topológica do grafo para agregar os vértices considerando a hierarquia dos rótulos. A partir dessas formas de análise, foram introduzidos dois tipos de medidas: I-aggregated graph, que lida com múltiplos valores das informações, e T-aggregated graph, que lida com a estrutura topológica do grafo. Além dessas definições, o artigo apresentou as definições das operações OLAP (por exemplo, roll-up, drill-down, slice

/ dice) no grafo agregado e medidas definidas.

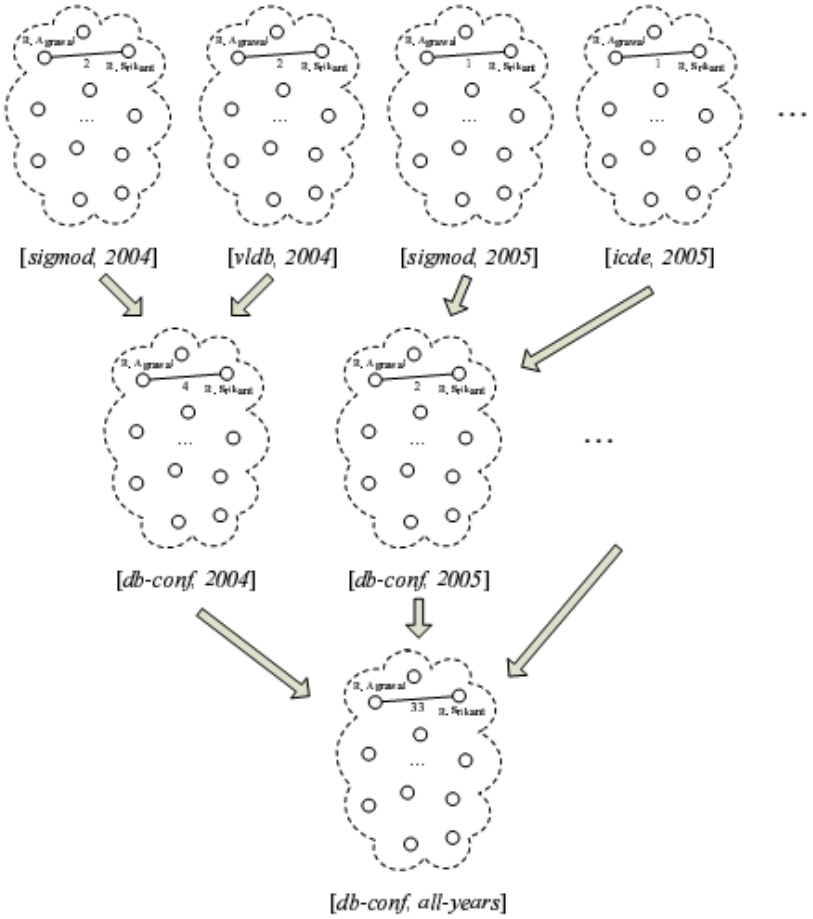


Figura 4 - Grafos com dimensões nos rótulos (CHEN et al., 2009)

Apesar da introdução do conceito T-OLAP, o trabalho enfatizou apenas as medidas do tipo I-OLAP, enquanto que as T-OLAP ficaram para trabalhos futuros. A Figura 5 apresenta um exemplo da medida I-OLAP, realizando a agregação de autores em razão da instituição

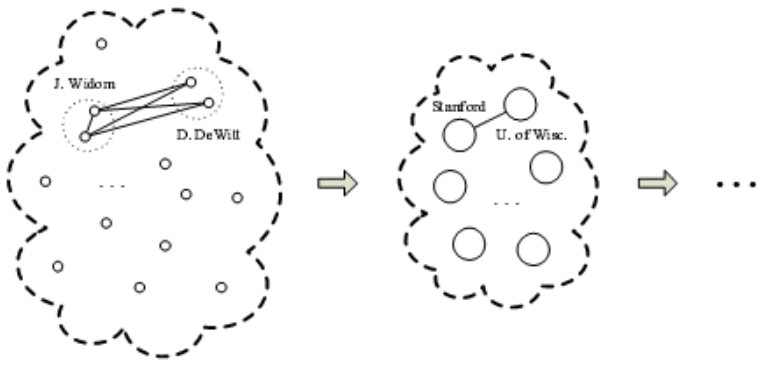


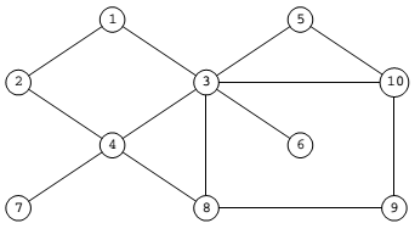
Figura 5 - Agregação do autores da mesma instituição (CHEN et al., 2009)

Na especificação das medidas do tipo I-OLAP foram definidas três categorias, considerando a dificuldade de processamento: *I-distributive*, que processam células de baixo nível para compor o resultado; *I-algebraic*, que combina medidas do tipo *I-distributive* com funções algébricas; e *I-holistic*, que representam as funções complexas de agregação em grafo. Contudo, esse trabalho introduziu vários conceitos importantes para área, sendo um dos trabalhos pioneiros nesse campo de pesquisa.

3.2.2 Graph Cube

No trabalho de ZHAO et al. (2011), os autores reconhecem a necessidade de aplicar a tecnologia OLAP em grafo e introduziram o conceito de *Graph Cube*, que combina as características de grafos multidimensionais com as tecnologias de cubo de dados. O *Graph Cube* é um modelo de *Data Warehouse* que essencialmente contém um conjunto de todas as possíveis agregações em um grafo multidimensional que fornece serviços de apoio à decisão. Este modelo considera os atributos do vértice como dimensões e os grafos agregados como resultado das medidas. Enquanto os sistemas OLAP tradicionais agregam as informações armazenadas no cubo utilizando valores numéricos, o *Graph* Cube retorna essa informação utilizando um grafo agregado, chamado cuboid, para representar os valores agregados.

O *Graph Cube* é obtido por meio da reestruturação do grafo multidimensional original em todas as possibilidades de grafo agregado, utilizando como base o conjunto de atributos. Um grafo agregado é um resumo do grafo original contendo uma ou mais dimensões. A sua modelagem resulta em um grafo com pesos, no qual os vértices são agregados em função dos valores das dimensões e seu peso é calculado com o uso de funções de agregação (COUNT, SUM, AVG). As arestas são agregadas e o seu peso é definido também pela função agregada.



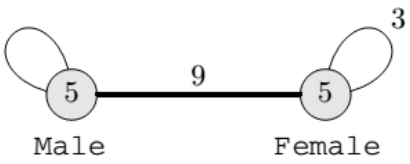
(a) Graph

ID	Gender	Location	Profession	Income
1	Male	CA	Teacher	$70,000
2	Female	WA	Teacher	$65,000
3	Female	CA	Engineer	$80,000
4	Female	NY	Teacher	$90,000
5	Male	IL	Lawyer	$80,000
6	Female	WA	Teacher	$90,000
7	Male	NY	Lawyer	$100,000
8	Male	IL	Engineer	$75,000
9	Female	CA	Lawyer	$120,000
10	Male	IL	Engineer	$95,000

(b) Vertex Attribute Table

Figura 6 - Grafo multidimensional (ZHAO et al., 2011)

Estes grafos agregados também podem ser chamados de cuboids. Um exemplo de um grafo multidimensional é mostrado na Figura 6, em que o grafo mostra os relacionamentos entre os vértices e a tabela mostra os atributos para cada vértice. A Figura 7 mostra um grafo agregado no qual os dados são agregados em função da dimensão gênero.



(a) Aggregate Network

Gender	COUNT(*)
Male	5
Female	5

(b) Aggregate Table

Figura 7 - Grafo agregado (ZHAO et al., 2011)

Além da modelagem dos dados, os autores propuseram duas formas de consulta OLAP:

- *Cuboid query* - os vértices agregados e as arestas atendem à dimensão solicitada na consulta e podem trabalhar com qualquer função de agregação (SUM, AVG, COUNT). Por exemplo, considere um grafo com os vértices contendo os atributos (gênero, localização, profissão). Um exemplo de consulta para este grafo pode ser (gênero, *, *). No resultado, todos os vértices que possuem o mesmo valor no atributo da dimensão gênero formam um vértice agregado e as arestas também são agregadas utilizando a função COUNT; e
- *Crossboid query* - é uma consulta que retorna um grafo agregado cruzando as informações entre duas ou mais estruturas de cuboid. Um exemplo de uma consulta *crossboid* pode ser a comparação da renda média de um usuário com ID= 3 com a média salarial em vários locais, considerando o gênero, como mostra a Figura 8. Apesar de ser natural comparar a renda do usuário 3 com a renda média dos

usuários em Illinois, essa comparação cruza as informações de cuboids nas consultas OLAP.

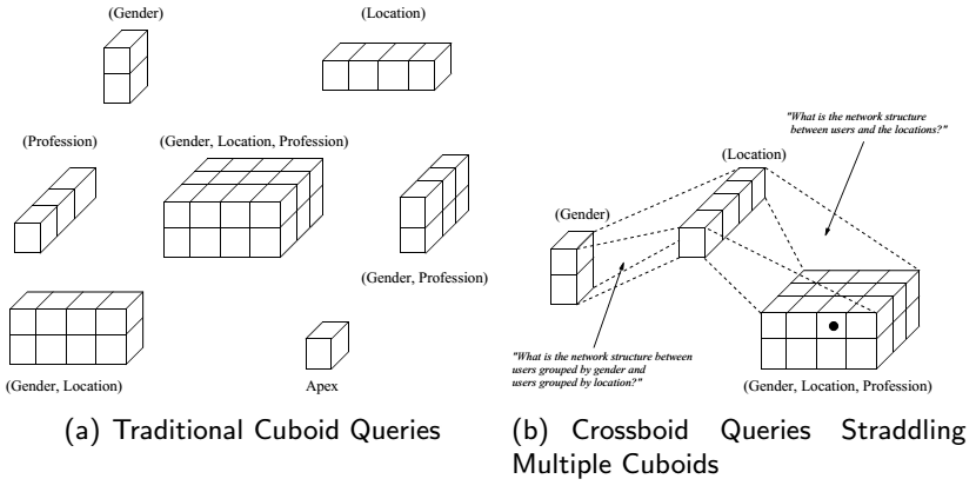


Figura 8 - Consulta Crossboid (ZHAO et al., 2011)

Em termos de implementação, a implementação do *Graph Cube* é definida por meio da criação de uma estrutura de *Graph Cube Lattice*, na qual cada vértice do grafo representa um cuboid no *Graph Cube*, como mostra a Figura 9. Considerando uma rede multidimensional com n dimensões, o *Graph Cube* correspondente terá 2^n cuboids. Dessa forma, os grafos agregados correspondentes em cada cuboid são computados e inseridos na estrutura. Os autores identificam três possibilidades de integrar os cuboids na estrutura de *Graph Cube Lattice*: *full materialization*, *no materialization* e *partial materialization*. As duas primeiras (naturalmente) ocupam muito espaço de dados e tempo de consulta, em qualquer grafo menos trivial. Com isso, o trabalho buscou a terceira possibilidade uma materialização parcial que atenda as consultas, as quais exigem um novo cuboid, por meio do processamento dos cuboids pré-processados.

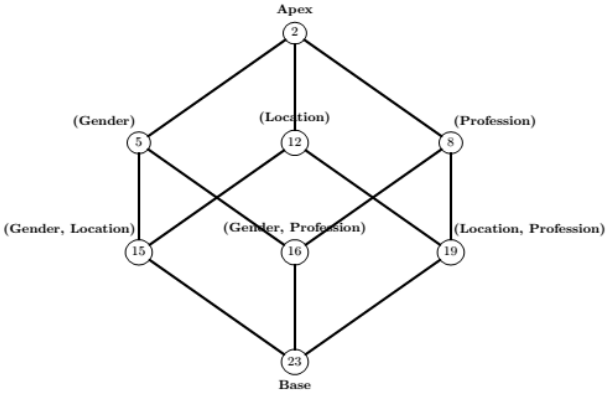
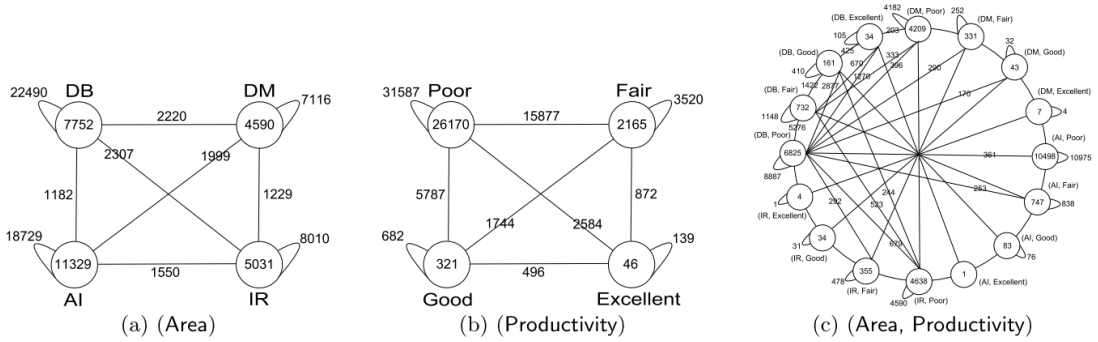


Figura 9 - Graph Cube Lattice (ZHAO et al., 2011)

Na parte de experimentação, o trabalho apresentou resultados de algumas consultas que utilizam um rede de colaboração entre autores. Essa rede de colaboração foi extraída do DBLP e modificada para incluir um atributo categórico que classifica a produtividade do autor, como mostra a Figura 10. Na ilustração da Figura 11 aparecem os resultados das consultas utilizando na dimensão da consulta os atributos de área e produtividade.

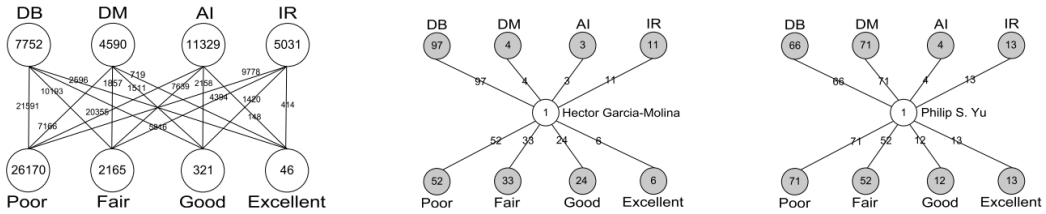
Productivity	Publication Number x
Excellent	$50 < x$
Good	$21 \leq x \leq 50$
Fair	$6 \leq x \leq 20$
Poor	$x \leq 5$

Figura 10 - Atributo de produtividade do autor (ZHAO et al., 2011)



(a) (Area) (b) (Productivity) (c) (Area, Productivity)

Figure 9: Cuboid Queries of the Graph Cube on DBLP Data Set



(a) Area ⋈ Productivity (b) Area ⋈ Base ⋈ Productivity for "Hector Garcia-Molina" (c) Area ⋈ Base ⋈ Productivity for "Philip S. Yu"

Figura 11 - Consultas no Graph Cube (ZHAO et al., 2011)

Considerando essas informações, o Graph Cube é um marco para a representação do grafo agregado, sendo o primeiro a introduzir um formato para representar a agregação do grafo de dados.

3.2.3 A Framework for Building OLAP Cubes on Graphs

No trabalho de GHRAB et al. (2015), os autores propuseram um framework para construir cubos OLAP em grafos de dados heterogêneos e analisar as propriedades topológicas do grafo. Eles ampliaram os recursos de análise em grafo, integrando o GRAD (*GRAph Database model*), em um modelo de banco de dados orientado a análise em grafo (GHRAB et al., 2013, 2014). O GRAD suporta nativamente a representação de hierarquias e a análise do conteúdo dos vértices. Essas características são utilizadas para oferecer

suporte a hierarquias de dimensão e criar cubos OLAP sobre o grafo. Assim, eles propuseram uma técnica para construir cubos OLAP com a capacidade de realizar uma análise eficaz em vários níveis / dimensões de seu grafo de dados.

Os autores iniciam o trabalho apresentando a base de dados movielens, publicada pelo grupo de pesquisa GroupLens, que representa um grafo de filmes, como mostra a Figura 12 (a) ilustrando um subgrafo do grafo de filmes. Também é apresentado um esquema multidimensional na Figura 12 (b), apresentando uma modelagem dimensional correspondente.

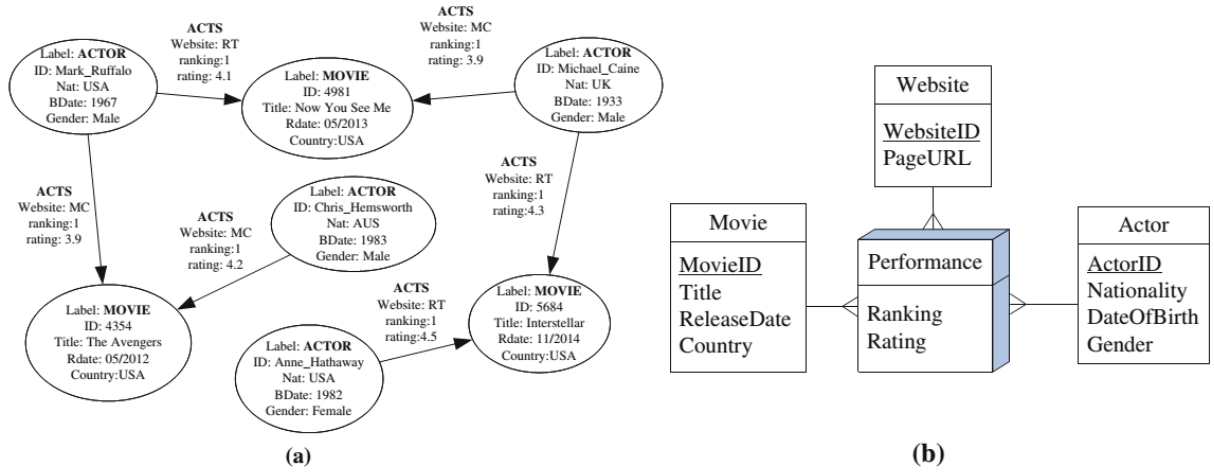


Figura 12 - Base de dados movielens (GHRAB et al., 2015)

O framework desenvolvido suporta dados multidimensionais em grafo de propriedades. Os elementos do grafo podem representar diferentes tipos de entidades e de relacionamentos entre essas entidades, representando assim os dados do mundo real. Além disso, permite o armazenamento de atributos tanto nos vértices como nas arestas do grafo. No trabalho, várias definições sobre os conceitos multidimensionais em grafos são definidas, tais como:

- *Dimension Level* - para especificar o nível de uma dimensão;
- *Dimension* - para compor uma dimensão utilizando um conjunto ordenado de *Dimension Level* ;
- *Measures* - para calcular as medidas do cubo podendo utilizar um tipo de algoritmo com uma função de agregação. Esse tipo de algoritmo caracteriza as *Measures* nas seguintes classificações:
    - *Content-Based Measures* - são medidas semelhantes às medidas tradicionais de DW, que utiliza apenas os atributos do grafo para o cálculo da medida, por exemplo, SUM, AVG, COUNT; e

    - *Graph-Specific Measures* - são medidas que capturam a propriedade topológica do grafo para o cálculo da medida, por exemplo, *degree centrality e closeness centrality*.
- *Aggregate Graph* é a definição de um grafo agregado com pesos em seus elementos, no qual os vértices agregados representam um conjunto de vértices do grafo de dados original e cada aresta agregada representa a fusão das arestas entre pares de vértices agregados; e
- *Graph Cube* - corresponde a um conjunto de grafos agregados, os quais podem ser chamados de cuboids, obtidos por meio de todas as possibilidade de agregação das informações do grafo original.

Após apresentadas as definições dos conceitos multidimensionais, o artigo apresenta o processo de construção do cubo OLAP sobre o grafo de propriedade, o qual descreve um multigrafo dirigido, rotulado e com atributos. Os autores consideram que cada vértice contém apenas um rótulo, introduzindo assim o conceito de *class* para representar um conjunto de vértices que compartilham o mesmo rótulo.

Na produção do *Graph Cube*, foi definido um grafo *lattice* para computar todas as agregações OLAP possíveis no grafo de dados, de modo que cada vértice do grafo lattice corresponde a um cuboid, o qual representa um grafo agregado. Os autores assumem que para produzir um cubo é preciso analisar as propriedades e os relacionamentos entre as entidades para definir uma modelagem mais adequada para os dados. Desse modo, os autores consideram que os atributos das arestas podem ser modelados como uma dimensão, por exemplo, o atributo Website da aresta ACTS apresentada no esquema multidimensional da Figura 12 (b). Entretanto, os atributos *ranking* e *rating* são considerados medidas dessa modelagem.

Para melhorar o entendimento na produção de um *Graph Cube*, os autores apresentaram um exemplo aplicando dimensões e medidas sobre o grafo de propriedade da Figura 12 (a) para produzir um grafo *lattice* com o da Figura 13.

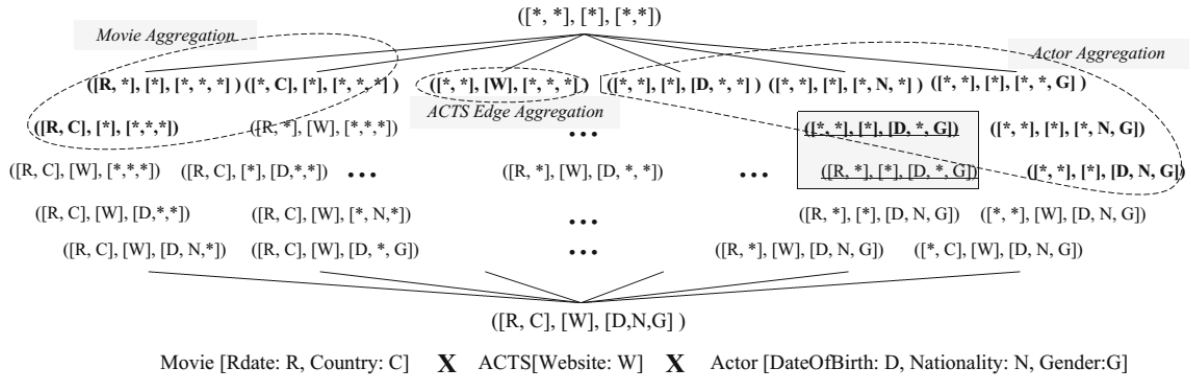


Figura 13 - Estrutura Lattice e cuboids (GHRAB et al., 2015)

Na Figura 14, eles apresentaram o grafo agregado dos filmes considerando a data de nascimento e o gênero dos atores. Além disso, eles perceberam que as medidas do tipo *Graph-Specific Measures* (e.g., *closeness centrality* de atores) não podem ser reusadas na mudança de nível da dimensão, pois em cada nível há uma topologia diferente no grafo para ser calculada.

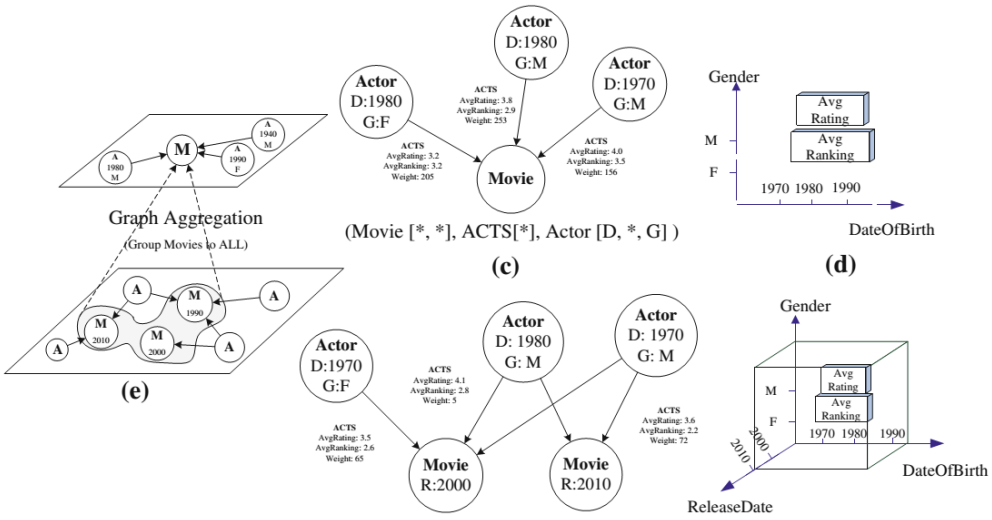


Figura 14 - Grafo agregado (GHRAB et al., 2015)

Os autores também apresentaram uma modelagem no grafo *lattice* que apresenta uma dimensão hierárquica com níveis de hierarquia. A representação dessas modelagens possibilita agregar os dados em grafo considerando os níveis da dimensão, como mostra o exemplo da Figura 15.

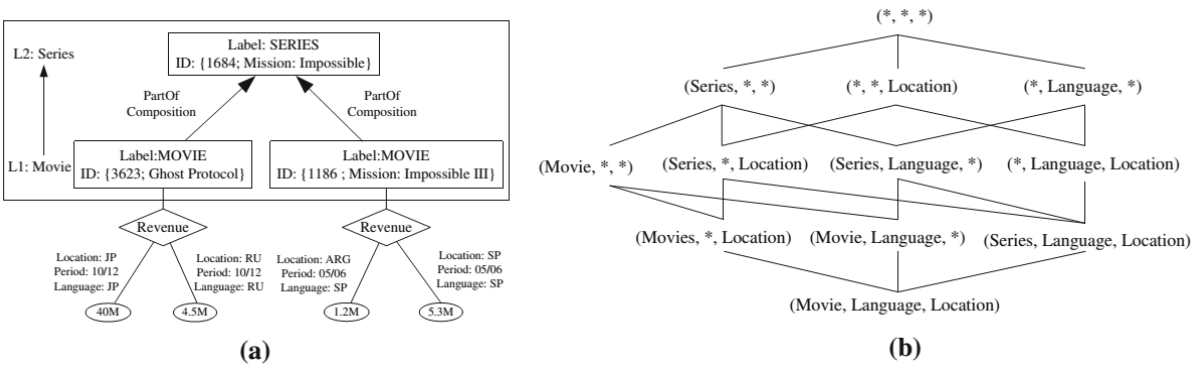


Figura 15 - Dimensão hierárquica em grafo lattice (GHRAB et al., 2015)

Com base nas definições, os autores apresentaram no final do artigo a especificação de um protótipo em Neo4j. A arquitetura dessa especificação é ilustrada na Figura 16 e os seus componentes são introduzidos, a seguir:

- *Graph ETL* - responsável por extrair dados de fontes de dados externas e importar no *framework*;
- *Graph storage and materialization* - responsável por armazenar os dados utilizando múltiplas instâncias do banco de dados Neo4j, podendo administrar uma instância para armazenar os dados do grafo e outra para armazenar o *lattice*;
- *Graph lookup and update* - responsável por integrar o banco de dados Neo4j com o Hadoop Distributed File System (HDFS), no intuito de preparar os dados para o processamento distribuído ou de agregação. Esse componente carrega os dados do Neo4j no HDFS para o processamento e, depois de terminado o processamento, armazena os dados do HDFS em uma nova instância do Neo4j; e
- *Graph Aggregation and Measures Computation* - responsável por produzir o cubo OLAP, processando os diferentes tipos de medidas em todos os cuboids para compor o grafo lattice. Nesse processamento é utilizada a biblioteca GraphX para melhorar o desempenho e os resultados são persistidos em uma instância do Neo4j.

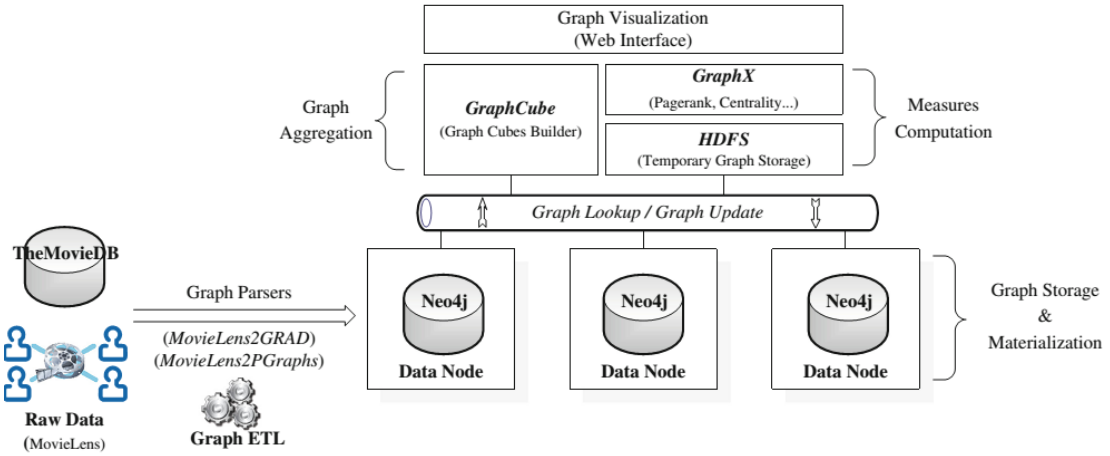


Figura 16 - Protótipo de implementação (GHRAB et al., 2015)

Com base na abordagem, esse trabalho apresentou um grande avanço em comparação aos outros. Utilizou um SGBDG que atende ao modelo de grafo de propriedade e apresentou medidas que combinam diferentes algoritmos de análise.

3.2.4 Iceberg Cube

Nos trabalhos de YIN, Dan et al. (2016); YIN, Dan e GAO (2014), os autores consideram que um sistema de rede de informação heterogêneo precisa conter tipos de vértices e arestas heterogêneos. Além disso, definem que um *meta-path* é a especificação de um caminho que conecta os vértices por meio de uma sequência heterogênea de arestas, representando assim diferentes tipos de relações semânticas entre os vértices. Outrossim, eles afirmam que os *meta-paths* são bons mecanismos para melhorar a qualidade da análise dos grafos em rede de informações heterogêneas. Em virtude disso, os autores apresentam, nesse artigo, uma estrutura de cubos de iceberg para redes de informações heterogêneas baseadas em *meta-path*. Nessa pesquisa, eles afirmam que não existem trabalhos na literatura sobre cubo de *iceberg* em grafo, o qual realiza o processamento dos cuboids em função das informações mais relevantes da base de dados que atendem ao *meta-path*.

Na explanação dos trabalhos, os autores adotaram uma base de dados do mundo real, chamada *IMDb network*[10], para ilustrar exemplos de redes de informações heterogêneas e *meta-paths*. Nessa base de dados ilustrada na Figura 17, há quatro tipos de vértices {*Movie* (M), *Actor* (A), *Director* (D) e *Movie Studio* (S)} e quatro tipos de arestas {*Cooperated* (A-A), Play (A-M), Direct (M-D) e Publish (M-S)}.

[10] http://www.imdb.com/

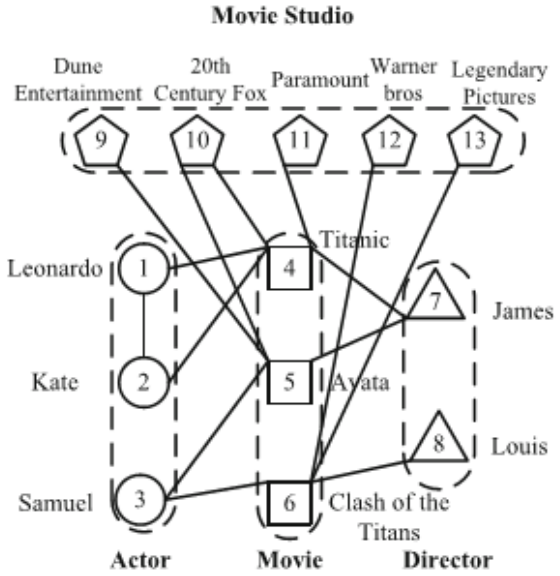


Figura 17 - Base de dados IMDb (YIN, Dan et al., 2016)

***Meta-path*:** Um meta-path P é denotado na forma de T_1-T_2 -. . . - T_{n+1}, que define uma relação composta entre os tipos T_1 e T_{n+1}. Já uma instância de *meta-path* é um caminho p = (a_1, a_2 . . . a_{n+1}) entre a_1 e a_{n+1} que segue as especificações do *meta-path* P. Além disso, os autores consideram a aplicação de *meta-path* simétrico na produção de cubo de iceberg. O meta-path simétrico é uma representação de um meta-path que conecta os mesmos tipos de vértices nas duas extremidades, por exemplo, um caminho de comprimento 2 atende ao padrão T_1-T_2-T_1 e o de comprimento 3 atende ao padrão T_1-T_2-T_3-T_2-T_1, representando assim meta-paths simétricos.

Com base nessas definições, os autores apresentam uma representação de *meta-path* A-M-D que denota a relação entre um ator e um diretor indicando quais diretores dirigiram o filme que os atores atuaram. Essa forma de delinear os dados usando *meta-path* é considerada pelos autores com um poderoso mecanismo para selecionar vértices. A Figura 18 apresenta dois cuboides da rede IMDb, os quais agregam os vértices de acordo com o *meta-path* especificado. Na Figura 18 (a), por exemplo, o *meta-path* simétrico A-M-A agrega os atores {1 e 2} que participaram no filme {4}. Já a Figura 18 (b) com o meta-path simétrico S-M-D-M-S agrega os vértices de estúdio {9, 10 e 11} e dos filmes {4 e 5} por meio do vértice de diretor {7}.

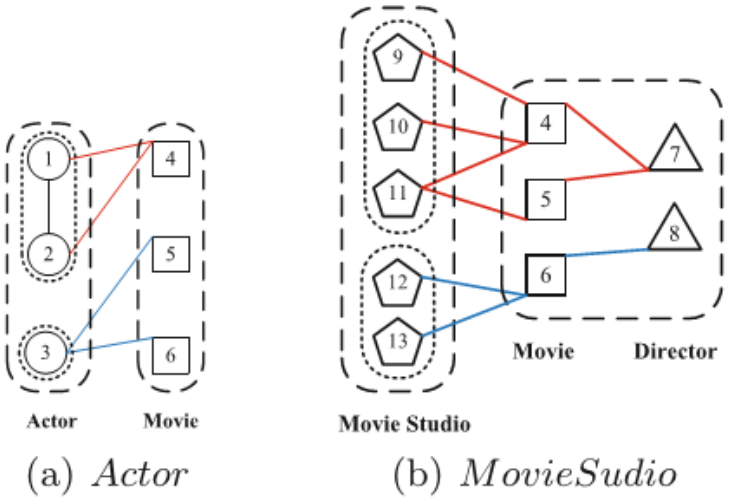


Figura 18 - Cuboids de rede IMDb (YIN, Dan et al., 2016)

Nessa pesquisa, os autores afirmam que a produção de cuboids sobre grandes *meta-paths* é explosiva, fazendo com que seja necessário limitar a quantidade de cuboids. Assim, a proposta de um cubo de iceberg é bem aceita, pois o cubo de iceberg pode restringir o número de cuboids limitando a produção aos dados de maior interesse.

Na produção de um cuboid, é necessário ter um *meta-path* simétrico, um valor de *threshold* e funções de agregação. Para a identificação dos dados que compõem o cuboid, é aplicada sobre o grafo de dados uma função de similaridade com o *threshold*, que foi definida pelos autores para identificar as instâncias de caminho comparáveis com o *meta-path* simétrico. Após esse processo, funções de agregação são aplicadas sobre os vértices dessas instâncias, no intuito de formar os cuboids, denominados *iceberg* cuboid.

Nesse trabalhos, a produção de um cubo de *iceberg* consiste em produzir todos os *iceberg* cuboids, de modo que esse problema é NP-hard. Consequentemente, esse processamento inviabiliza a produção do cubo de *iceberg*. Dessa maneira, os autores apresentam uma solução heurística, que utiliza o algoritmo *slice tree*, para reduzir o espaço de busca das instâncias dos caminhos e, assim, produzir um cubo de *iceberg* que atenda às especificações dos *meta-paths*. Além disso, o trabalho não menciona a materialização dos dados, fazendo com que o cubo de iceberg seja executado durante as consultas, usando o meta-path e o *threshold* para restringir os dados de análise.

Diante do exposto, esses trabalhos apresentaram uma forma diferente de analisar o grafo utilizando *meta-path e threshold*. Essa abordagem se distancia da forma tradicional, pois não utilizou os conceitos básicos de cubo de dados, tais como dimensões, medidas e fato. Com isso, consideramos que os trabalhos não empregam de fato o cubo de dados, mas uma forma de consulta agregada em grafo que foi baseada em “Iceberg Queries”

(FANG et al., 1998; NAN LI et al., 2013). Contudo, esses autores foram os pioneiros em utilizar padrão de caminho (meta-path) para analisar dados em grafo, mostrando uma forma diferente de análise.

3.2.5 EvOLAP Graph

No trabalho de GUMINSKA e ZAWADZKA (2018), os autores consideram que a combinação das tecnologias OLAP com os Sistemas Gerenciadores de Bancos de Dados em Grafos (SGBDG) pode resultar em melhores análises com decisões mais precisas. No entanto, nos data warehouses tradicionais, a informação topológica das relações entre os dados é definida durante a produção da modelagem dimensional e, portanto, não pode ser simplesmente adaptada à natureza flexível do grafo.

A análise de histórico é uma característica dos data warehouses e pode ser aplicada ao SGBDG, mesmo que o grafo de dados, com características dinâmicas e heterogêneas, não possua um esquema de dados. Com isso, qualquer alteração da informação do grafo repercute em mudanças na estrutura do grafo, o que leva a alterar os dados históricos. Dessa forma, os autores propuseram um modelo em grafo para rastrear as alterações das informações topológicas e permitir a realização de consultas analíticas OLAP.

Esse modelo, baseado no modelo de grafo de propriedades, integra os recursos da tecnologia OLAP e estabelece um rastreamento das alterações das informações topológicas em função do tempo (VIJITBENJARONK et al., 2017), no intuito de permitir analisar o histórico em um ambiente dinâmico como o grafo.

Para realizar o rastreamento, a abordagem EvOLAP Graph utiliza um modelo que divide a informação de uma entidade em uma estrutura de hipervértice, a qual contém um único vértice de identidade (Vi), que identifica a entidade e a localiza na topologia do grafo, e um conjunto de vértices de estado (Ve), que registra as mudanças das informações da entidade em função do tempo. Nessa estrutura em hipervértice, as arestas possuem os atributos “From” e “To” para indicar o período de validade da informação no hipervértice.

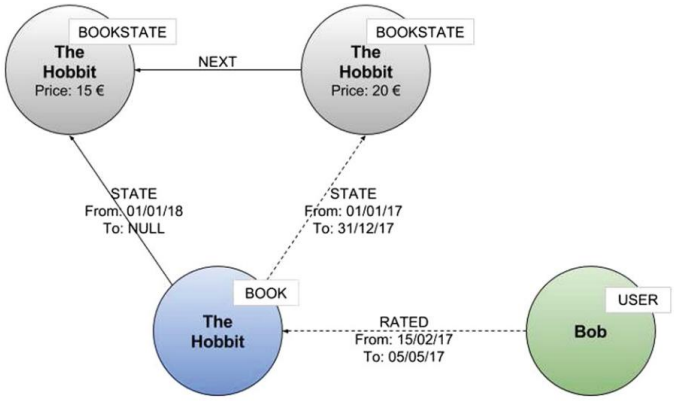


Figura 19 - Modelo com versionamento do grafo (GUMINSKA e ZAWADZKA, 2018)

A Figura 19 mostra um exemplo de versionamento do grafo. Nesse exemplo, o Vi “Book” que representa o livro “The Hobbit” é versionado, de modo que o livro no intervalo de “01/01/2017” à “31/12/2017” tinha o preço de 20 euros no Ve “BookState” e depois de “01/01/2018” ficou com o preço de 15 euros em um novo Ve. Nessa ilustração, os relacionamentos tracejados indicam que o seu intervalo de tempo não atende à data “01/01/2018”. Com isso, esse modelo de controle de versão fica encarregado de registrar as mudanças e evitar a deleção dos dados, registrando o intervalo de tempo para validar a informação.

Com a definição dessa estrutura de controle de versão, os autores especificaram os principais conceitos do data warehouse - dimensões, medidas, fatos e hierarquias - no EvOLAP Graph. Na representação desses conceitos, eles consideram que os elementos do grafo – vértice, aresta e propriedade – podem ser definidos como dimensões, medidas ou fatos apenas durante o tempo de vida da consulta.

Dimension. A dimensão corresponde a uma entidade do grafo. No EvOLAP Graph, uma instância de uma entidade pode ser representada por um hipervértice, contendo vértice de identidade (Vi) e vértices de estado (Ve). Desse modo, a dimensão pode ser representada por vértice e hipervértice. A Figura 20 mostra uma parte do grafo contendo vértices com quatro rótulos diferentes: “Book”, “Author”, “User” e “Bookstate”. Dessa parte do grafo, os autores distinguiram três dimensões: autores, livros e usuários. Com essa distinção, o hipervértice do livro “The Hobbit” é considerado uma instância da dimensão livros da mesma maneira que o vértice “Mr. Bliss”, como mostra a Figura 20.

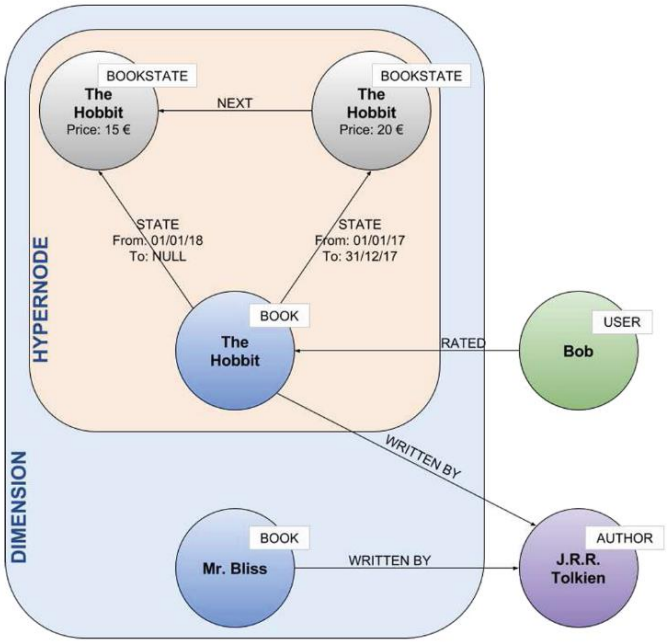


Figura 20 - Definição da dimensão no EvOLAP Graph (GUMINSKA e ZAWADZKA, 2018)

Measure. A medida representa uma métrica usada para analisar um assunto. No EvOLAP Graph, são considerados dois tipos de medidas: Informational Measure, para processar valores numéricos sobre as propriedades dos elementos do grafo, e Topological Measure, para processar informações da estrutura topológica do grafo.

Fact. No EvOLAP Graph não existe um representação explícita do fato no grafo de dados. O fato pode ser representado por um vértice ou aresta, dependendo da consulta. Dessa forma, os autores consideram que o fato está oculto e distribuído em todo o grafo.

Hierarchy. Nesse modelo a representação hierárquica da dimensão é definida por meio de arestas hierárquicas que relacionam as instâncias das dimensões, como mostra a Figura 21, que define uma hierarquia sobre a entidade livro.

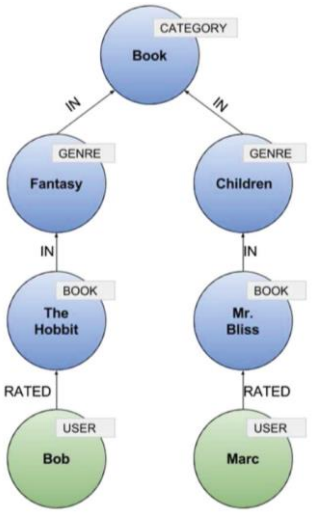


Figura 21 - Hierarquia no EvOLAP Graph (GUMINSKA e ZAWADZKA, 2018)

No final do artigo, os autores apresentam consultas em linguagem natural e gráficos com resultados do desenvolvimento dessa consulta, não exemplificando a linguagem de consulta ou o formato da resposta. Além disso, as consultas em linguagem natural não abordam operadores OLAP e não expressam como é realizado o processo de agregação.

Diante do exposto, consideramos que esse trabalho explora uma tendência, na qual utiliza os SGBDG para realizar a análise OLAP do grafo de dados. Define uma forma para preservar os dados históricos no grafo de dados, entretanto, esse artigo não detalhou o processo de consulta e análise dos dados em grafo.

3.3 ANÁLISE COMPARATIVA

Neste capítulo, introduzimos importantes trabalhos em uma sequência cronológica de publicações. Nessa sequência é possível identificar a evolução das propostas e as principais características. Para facilitar o entendimento, apresentamos na Quadro 2 um breve resumo de cada trabalho destacando as suas principais contribuições.

Quadro 2 - Contribuições dos artigos

Referência	Contribuições
Graph OLAP (CHEN et al., 2008, 2009)	Os autores propuseram uma abordagem para agregar diferentes grafos que contêm informações correlatas a fim de aplicar OLAP em grafo. Os grafos são homogêneos e as dimensões são definidas nos rótulos. Definiram medidas que tanto podem utilizar os atributos quanto a topologia dos vértices, mas apenas detalharam as medidas que

	processam os atributos. Mostraram o processo de agregação, mas não apresentaram um grafo agregado. Utilizaram dados estáticos e não ofereceram soluções para lidar com alterações dos dados.
Graph Cube (ZHAO et al., 2011)	Os autores propuseram uma abordagem para produzir um cubo com os dados de um grafo homogêneo. As dimensões consideradas são as propriedades dos vértices e na estrutura do grafo as arestas não possuem propriedades ou rótulos. Apresentaram formas de realizar consultas que cruzam as informações dos cuboids. Utilizaram dados estáticos e não ofereceram soluções para lidar com as alterações dos dados. Foram os primeiros a representar uma estrutura de grafo agregado. Apresentaram medidas tradicionais de agregação e não integraram algoritmos de análise topológica na solução.
A Framework for Building OLAP Cubes on Graphs (GHRAB et al., 2015)	Os autores propuseram uma abordagem que produz um cubo no SGBDG com os dados de um grafo heterogêneo. Essa abordagem atende ao modelo de grafo de propriedade, permitindo dados heterogêneos com propriedades e rótulos nos elementos do grafo. As dimensões podem ser representadas por vértices ou propriedades das arestas. As medidas podem processar tanto os valores das propriedades quanto a topologia dos vértices, além de possibilitar a combinação de diferentes algoritmos de análise. O cubo consiste em um grafo *Lattice* constituído por todas as possibilidades de resposta em grafos agregados. Apresentou um protótipo de implementação que utiliza o Neo4j para materializar o cubo e uma arquitetura distribuída para processar esse cubo. Na abordagem foi integrada apenas a operação OLAP roll-up e não existe uma proposta para lidar com alterações do grafo de dados, tendo que produzir um novo cubo em caso de alterações nos dados.

Iceberg Cube (YIN, Dan et al., 2016; YIN, Dan e GAO, 2014)	Os autores propuseram uma abordagem que produz um cubo de iceberg com os dados de um grafo heterogêneo. Essa abordagem se distancia da análise OLAP tradicional, visto que não definiu as estruturas básicas do cubo OLAP, tais como dimensões e fato. Com isso, os autores foram pioneiros ao utilizar padrões de caminho (meta-path) para analisar e agregar os dados do grafo. Essa abordagem não realiza materialização, podendo se adaptar facilmente às alterações dos dados. Os autores apresentam medidas que realizam funções de agregações tradicionais, não abordando formas de análises topológicas. Os autores mencionam a utilização do grafo agregado, mas não apresentam uma representação.
EvOLAP Graph (GUMINSKA e ZAWADZKA, 2018)	Os autores propuseram uma abordagem que define um modelo em grafo para estruturar o armazenamento de um grafo de propriedade heterogêneo no SGBDG. Esse modelo realiza um controle de versão sobre as informações topológicas do grafo para permitir análises de dados históricos. Na especificação da abordagem, os autores consideram que os conceitos do data warehouse - dimensões, medidas, fatos e hierarquias – só são definidos durante a realização da consulta, ou seja, o modelo não estabelece uma estrutura explícita para esses conceitos, tendo que representá-los durante a consulta. Nessa abordagem, as dimensões são entidades que são representadas por vértices e hipervértices, e podem ter relacionamentos hierárquicos para permitir operações OLAP - drill-down e roll-up. As medidas podem processar tanto os valores das propriedades quanto a topologia dos vértices. Além disso, o trabalho não menciona grafo agregado ou análises de padrões em grafo.

Com base nessas contribuições e nas contribuições desenvolvidas nesta tese, destacamos as principais características, definindo uma nomenclatura e uma descrição para cada característica:

- **Análise topológica -** consiste em utilizar algoritmos para analisar a estrutura topológica do grafo. Essa característica utiliza algoritmos de análise de redes e de teoria dos grafos para processar a estrutura topológica, por exemplo: *degree centrality*, *betweenness centrality*, *closeness centrality* (BORGATTI e EVERETT, 2006; FREEMAN, 1978; NEWMAN, 2010);
- **Análise OLAP -** consiste em utilizar múltiplas dimensões e operadores OLAP - slice, dice, roll-up e drill-down - para produzir um resultado agregado;
- **Análise topológica OLAP** - consiste em combinar as duas formas de análise utilizando as tecnologias OLAP para agregar os resultados da análise topológica;
- **Análise de padrões** - consiste em utilizar padrões em grafo, tais como meta-path, para realizar a análise do grafo de dados;
- **Grafo agregado** - consiste em produzir respostas de consulta que expressem tanto os valores agregados da propriedade do grafo, quanto a representação estrutural do grafo agregado;
- **SGBDG** - consiste em utilizar os recursos de um SGDBG na abordagem;
- **Materialização** – consiste em pré-processar os dados do grafo para lidar com grafo de dados volumosos, melhorando assim o tempo de resposta das consultas;
- **Atualização** - consiste em suportar mudanças no grafo de dados sem necessitar de um pré-processamento para a realização de consultas.

Com a definição dessas características, apresentamos na Quadro 3 uma avaliação sobre a cobertura dessas características nos trabalhos apresentados. Para essa avaliação, definimos as seguintes nomenclaturas: ✓ aborda a característica detalhando o seu funcionamento; ✗ não aborda a característica; e ⚠ menciona ou define a característica, mas não cumpre totalmente a característica.

Quadro 3 - Comparando as características dos trabalhos

Trabalho	Análise top.	Análise OLAP	Análise top. OLAP	Análise de padrões	Grafo agregado	SGBDG	Materialização	Atualização
Graph OLAP	⚠	✓	✗	✗	⚠	✗	⚠	✗
Graph Cube	✗	✓	✗	✗	✓	✗	✓	✗

Framework OLAP Cubes								
Iceberg Cube								
EvOLAP Graph								

Constatamos na revisão sistemática que após o ano de 2010, com os lançamentos dos SGBDG no mercado, muitos trabalhos passaram a integrar SGBDG em suas abordagens (BACHMAN, 2013; CASTELLTORT e LAURENT, 2014; GHRAB et al., 2015; GUMINSKA e ZAWADZKA, 2018; JAKAWAT et al., 2016b; LIU e VITOLO, 2013), levando a utilizar os recursos do SGBDG não só para armazenar os dados do grafo, como o trabalho de GHRAB et al. (2015), mas também para analisar os dados, como o trabalho de GUMINSKA e ZAWADZKA (2018). Os trabalhos mais recentes vêm utilizando grafos heterogêneos contendo propriedades e rótulos nos elementos do grafo, ou seja, aceitam o modelo de grafo de propriedade para manipular os dados. Além disso, identificamos poucos trabalhos que combinam a tecnologia OLAP com os algoritmos de análise em grafo, de modo a utilizar as duas formas de análise na mesma consulta, como introduzido no trabalho de GHRAB et al. (2015).

Considerando a Quadro 3, constatamos que as abordagens cujas análises são realizadas durante a consulta têm apresentado maior adaptação às mudanças. Esse fato é justificável em razão dessas abordagens não precisarem reestruturar os dados para permitir as análises em caso de mudanças, visto que os dados podem ser estruturados durante a inserção ou na realização das consultas. Nessa avaliação, percebemos que a maioria dos autores aceita a estrutura do grafo agregado para representar a resposta da agregação do grafo de dados.

Analisando esses trabalhos, inferimos três tendências de pesquisa. A primeira consiste em exprimir a informação agregada dos grafos, combinando a estrutura topológica e os valores das propriedades em uma representação de grafo agregado. A segunda consiste em desenvolver modelos e recursos para serem incorporados aos SGBDG, de modo que as tecnologias OLAP e os algoritmos de análise em grafo possam ser usados nas consultas do SGBDG para realizar análises em grandes grafos de dados. A terceira consiste em utilizar padrões em grafo para analisar tanto a topologia do grafo quanto a interação desses padrões no grafo de dados.

Observando os trabalhos, notamos que muitos utilizam a representação de grafo agregado para retratar as respostas das consultas OLAP em grafo, tais como (DENIS et

al., 2013; GHRAB et al., 2015; WANG, Pengsen et al., 2015; WANG, Zhengkui et al., 2014; ZHAO et al., 2011). Dessa forma, a primeira tendência pode ser considerada um requisito da área, visto que as consultas OLAP precisam lidar com a agregação da informação do grafo, a qual é constituída tanto pelos valores das propriedades quanto pela estrutura topológica do grafo. Apesar dessa tendência, não encontramos na revisão sistemática a especificação de uma interface gráfica que representasse de forma unificada a agregação dos valores das propriedades e da topologia dos grafos.

O trabalho de GHRAB et al. (2018) reafirma a segunda tendência mostrando a necessidade de um sistema de BI (*Business Intelligence*) que estenda os seus recursos para analisar os dados em grafo. Esse tipo de sistema, denominado "*Graph BI*", integra as tecnologias do data warehouse, do OLAP e da mineração do grafo para auxiliar a tomada de decisão em grafos de dados. Deste modo, a integração do armazenamento, a análise topológica OLAP e a visualização dos resultados são objetivos a serem alcançados nesse campo de pesquisa, sendo necessário o uso de um SGBDG para gerenciar e manipular um grande volume de dados em grafo.

Nas pesquisas recentes, o uso de *meta-path* para analisar grafo de dados é evidenciado em vários trabalhos (JAKAWAT et al., 2016a; YIN, Dan et al., 2016; YIN, Dan e GAO, 2014; ZHANG et al., 2017). Desse modo, a terceira tendência também ganha repercussão e passa a ser considerada um recurso importante na análise em grafo.

Com base nessas tendências, definimos uma abordagem que adiciona funcionalidades ao SGBDG, de modo que permita realizar consultas multidimensionais em grafo. Essas consultas integram a tecnologia OLAP e os algoritmos de análise em rede, de modo que possibilitem analisar padrões e relacionamentos do grafo de dados. Além disso, a abordagem desenvolvida mostra resultados de consultas que representam visualmente a agregação dos valores e das estruturas topológicas de um grafo de dados, que corresponde a um diferencial não contemplado nos trabalhos encontrados na revisão sistemática.

3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, mostramos o processo da revisão sistemática, detalhando a metodologia e as palavras-chave utilizadas na busca e seleção dos trabalhos. Em seguida, detalhamos os principais trabalhos com características que se aproximam dos recursos contemplados na nossa abordagem. Realizamos uma análise comparativa entre as características dos trabalhos detalhados e evidenciamos tendências de pesquisa com base

na pesquisa realizada. Assim, conseguimos descrever os principais trabalhos e as características que direcionam a pesquisa neste campo de conhecimento. Na análise comparativa, observamos que os trabalhos não contemplam formas de análise que combinem algoritmos topológica com a tecnologia OLAP para analisar padrões de grafo. Dessa forma, propusemos uma abordagem que combinem essas formas de análise e permita representar os resultados das consultas em grafos agregados. No capítulo a seguir, introduzimos essa abordagem de análise e detalhamos o modelo multidimensional em padrões de grafo que definimos para o uso junto ao SGBDG.

4 MODELO MULTIDIMENSIONAL EM GRAFO

A capacidade de analisar os dados é determinante para a descoberta de conhecimento e tomada de decisão. A possibilidade de mesclar diferentes métodos de análise na mesma consulta contribui com a extração de conhecimento, permitindo novas formas de análise. Considerando esse cenário, definimos nesta tese uma abordagem de análise em grafo que integra recursos de análise aos Sistemas Gerenciadores de Bancos de Dados em Grafos (SGBDG). Esses recursos são compostos por tecnologia OLAP e algoritmos de análise em grafo, de modo que permitam diferentes formas de análise no processamento de consultas em grafo. Para essa finalidade, definimos o Modelo Multidimensional em Padrões de Grafo (Modelo MPGrafo) que especifica a forma de modelar os dados no SGBDG, estruturando o grafo de dados em uma modelagem multidimensional orientada a assunto.

Este capítulo cobre o seguinte conteúdo. A Seção 4.1 apresenta uma visão geral da abordagem e a fonte de dados utilizada na explanação deste trabalho. A Seção 4.2 apresenta uma introdução do Modelo Multidimensional em Hipervértices. A Seção 4.3 define os conceitos e a aplicação do Modelo Multidimensional em Padrões de Grafo. A Seção 4.4 discute o uso do Modelo MPGrafo na modelagem dos dados. Por fim, a Seção 4.5 conclui o capítulo com as considerações finais.

4.1 VISÃO GERAL

Os ambientes que estamos considerando em nosso trabalho são caracterizados por possuírem dados heterogêneos com grande interconectividade que representam dados históricos de redes de informações. Esses dados geralmente são modelados em grafos para representar estruturas complexas de relacionamento. Para lidar com as características topológicas desses dados, os SGBDG têm sido essenciais no armazenamento e processamento dos dados, disponibilizando diferentes recursos para recuperar e analisar o grafo de dados. Nesta perspectiva, o ponto crucial que abordamos é como explorar os recursos de um SGBDG integrando a tecnologia OLAP e os algoritmos de análise em grafo, de forma que a abordagem possa realizar análises multidimensionais em padrões de grafo apresentando resultados em grafos agregados.

Nesse intuito, consideramos alguns aspectos para o desenvolvimento dessa abordagem. O primeiro consiste em definir um modelo multidimensional em grafo para organizar os dados no SGBDG de forma a facilitar o processamento de consultas

agregadas e multidimensionais em padrões de grafo - tal como, meta-path (YIN, Dan et al., 2016; YIN, Dan e GAO, 2014; YIN, Mu et al., 2012). Esse modelo precisa estar em consonância com as linguagens de consulta em grafo para estabelecer um padrão de consulta replicável em diferentes SGBDG. Além disso, o modelo precisa se adequar a diferentes SGBDG, podendo modelar o grafo de dados sem alterar as informações originais do mesmo.

O segundo aspecto é definir métodos para adicionar recursos OLAP e algoritmos de análise de rede nas consultas em grafo, permitindo assim consultas agregadas com análises topológicas e analíticas em grafo. Para isso, o modelo MPGrafo estabelece uma estrutura orientada a assunto para viabilizar a seleção de múltiplas perspectivas (dimensões), realizar operações OLAP e analisar a topologia do grafo de dados nas consultas. Com isso, torna-se possível combinar as diferentes formas de análise na mesma consulta.

Outro aspecto a ser considerado é a visualização dos resultados das consultas, para mostrar a informação das propriedades e da topologia do grafo de forma resumida. Na literatura, o grafo agregado é uma forma de visualização bastante difundida nos sistemas de grafos OLAP, sendo utilizado na representação e pré-processamento de consultas. Neste trabalho, adotamos essa forma de representação para retratar as respostas das consultas, visto que resume a informação do grafo de dados, levando em consideração a estrutura topológica e os valores das propriedades do grafo.

Diante desses aspectos levantados, apresentamos a Abordagem de Análise Multidimensional em Padrões de Grafo (AAMPGrafo), que adiciona características OLAP e algoritmos de análise em grafo aos recursos de um SGBDG. Essa abordagem permite ao usuário, analisar padrões em grafo, combinar diferentes análises na mesma consulta e visualizar os resultados no formato de grafos agregados.

A combinação de diferentes formas de análise na mesma consulta aumenta a complexidade de interação e de processamento dos recursos de análise. A AAMPGrafo consiste em adaptar o SGBDG para possibilitar o processamento de consultas multidimensionais e a visualização de grafos agregados. Com base nessas funcionalidades, separamos a abordagem em duas partes: Modelagem Multidimensional em Padrões de Grafo (Modelagem MPGrafo) e Consulta Multidimensional em Padrões de Grafo (CMPGrafo).

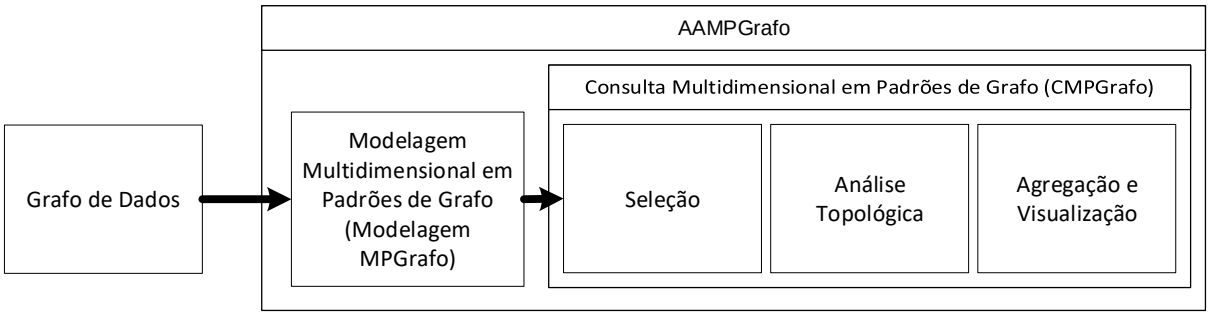


Figura 22 - Abordagem de Análise Multidimensional em Padrões de Grafo (AAMPGrafo)

A Figura 22 ilustra a AAMPGrafo indicando a sequência das etapas para realizar a análise dos dados em grafo. A Modelagem MPGrafo consiste em definir uma modelagem em grafo, utilizando o modelo de grafo de propriedades proposto (Modelo MPGrafo), que definimos a seguir. O CMPGrafo determina a sequência de processamento da consulta, subdividindo o processamento em etapas que são encarregadas de tarefas específicas: a seleção é encarregada de utilizar os recursos do SGBDG para recuperar os dados consultados; a análise topológica possibilita o processamento de algoritmos de análise em grafo nos dados recuperados; e a agregação e visualização realizam a agregação da estrutura topológica e dos valores das propriedades do grafo, no intuito de produzir a visualização do grafo agregado que responda à consulta solicitada. Para facilitar a compreensão dessa abordagem, utilizamos a mesma amostra de dados em todos os exemplos deste trabalho.

4.1.1 Fonte de dados adotada

A amostra de dados foi extraída de um repositório de dados aberto, denominado DBLP[11], e acrescida de outros dados para permitir a representação dos conceitos propostos e respectivos exemplos.

[11] https://dblp.uni-trier.de/

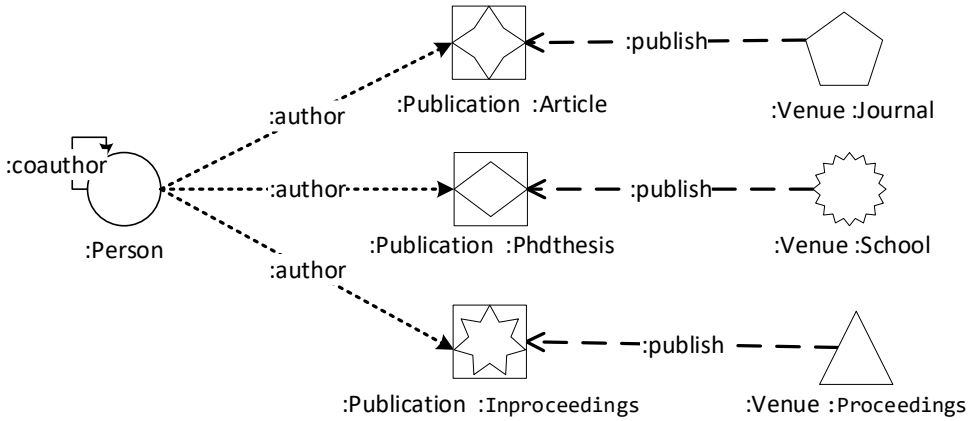


Figura 23 - Esquema em grafo dos dados do DBLP

Na Figura 23, apresentamos o esquema que representa os dados da amostra. Esse esquema mostra os diferentes tipos de objetos que compõem o grafo de dados. Os tipos de objetos são representados pelos rótulos de vértice (Rv) {:Person, :Article, :Phdthesis, :Inproceedings, :Journal, :School, :Proceedings } e pelos rótulos de arestas Ra {:coauthor, :author, :publish}. Na representação dos vértices, os autores possuem o rótulo (:Person), as publicações possuem dois rótulos, :Publication e {:Article ou :Phdthesis ou :Inproceedings} e os locais de publicação possuem dois rótulos, :Venue e {:Journal ou :School ou :Proceedings}. Já na representação das arestas, :coauthor representa os relacionamentos entre autores (:Person); :author os relacionamentos entre autor (:Person) e publicação (:Publication); e :publish o relacionamento entre publicação (:Publication) e local de publicação (:Venue).

Considerando as diferentes formas de se analisar os dados em grafo, extraímos uma pequena amostra do DBLP, que permita a representação e a visualização de diferentes modelagens orientadas a assunto.

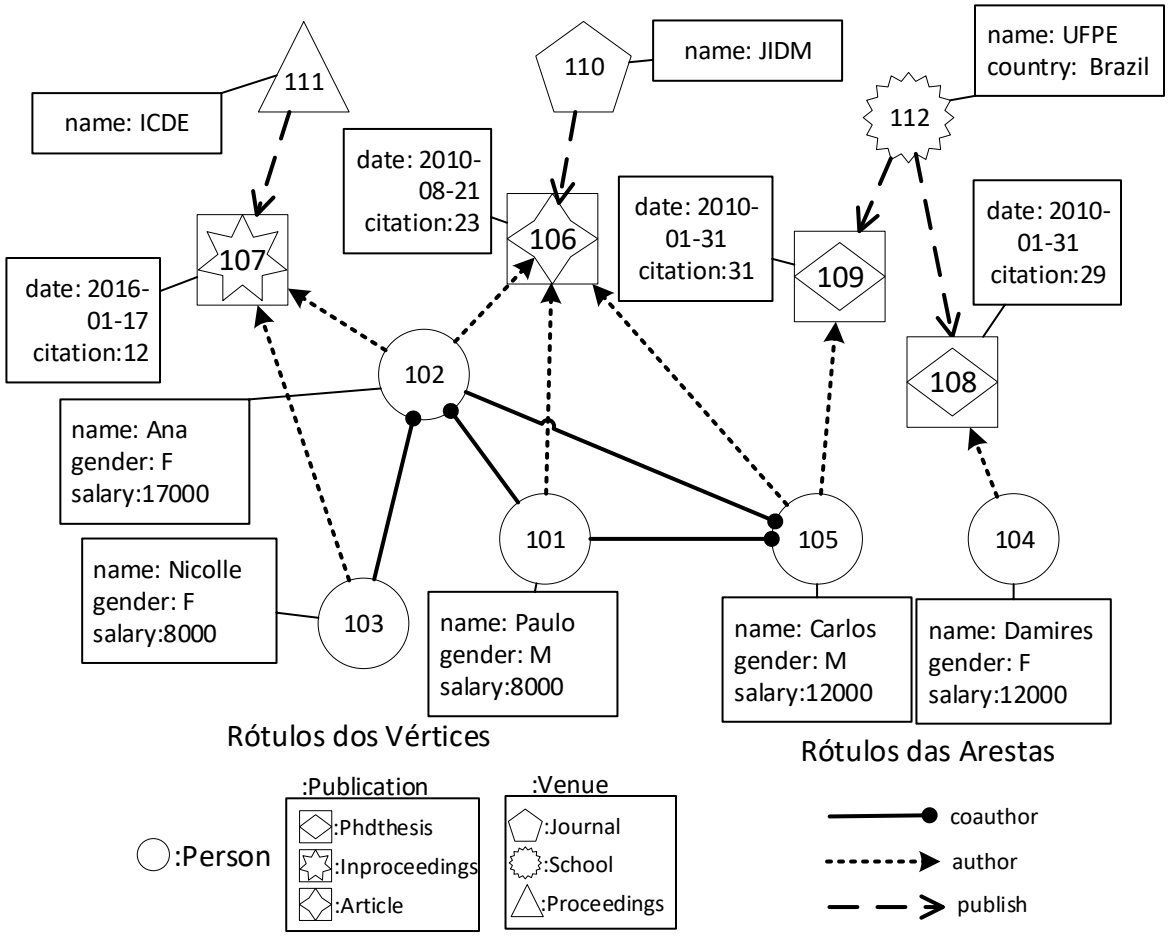


Figura 24 - Amostra de dados do DBLP

Na Figura 24, ilustramos a amostra de dados extraída do DBLP, contendo cinco autores, quatro publicações e quatro locais de publicações. Nessa amostra, é possível encontrar relacionamentos entre vértices do mesmo rótulo ou de rótulos diferentes, permitindo diferentes formas de modelagem. Nas próximas seções detalharemos alguns conceitos que retratam o modelo multidimensional.

4.1.2 Principais conceitos adotados

A modelagem multidimensional em grafo de propriedade é uma adaptação da modelagem dimensional definida por KIMBALL e ROSS (2002). Nessa modelagem existem três pilares essenciais para estruturar os dados: os fatos, as dimensões e as medidas. Os fatos são a menor unidade de informação que se deseja analisar, ou seja, o registro da menor granularidade de informação para a análise. As dimensões são perspectivas de análise que caracterizam as informações contidas no fato, distinguindo os fatos por meio das características que foram destacadas nas dimensões. As medidas são definições de cálculos que especificam como processar as informações dos fatos na consulta.

Na modelagem, as dimensões são criadas com base nas características e descrições dos fatos e cada fato se relaciona com as dimensões em função dos seus dados. A Figura 25 apresenta a ilustração que utilizaremos para representar o Modelo Estrela, na qual os hexágonos representam as dimensões e o círculo tracejado representa o conjunto dos fatos (KIMBALL e ROSS, 2002b).

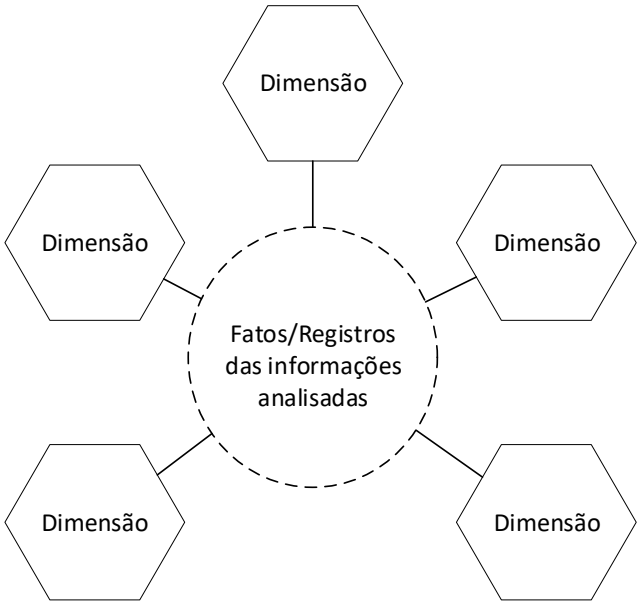


Figura 25 - Representação do Modelo Estrela

Observando a natureza do grafo, constatamos a importância de sua estrutura topológica. Dessa forma, consideramos que uma modelagem multidimensional em um grafo de dados precisa preservar duas informações importantes: as provenientes das propriedades contidas nos elementos do grafo e as correspondentes a estrutura topológica do grafo (tais como vértices, arestas e rótulos).

Consideramos que as informações do fato (foco da análise) em uma base de dados em grafo não podem se restringir apenas aos valores das propriedades, mas considerar também a estrutura topológica do fato na base de dados. Para isso, definimos inicialmente um modelo baseado em hipervértices que possibilita inserir a estrutura topológica do grafo de dados em um vértice, de modo a permitir análises multidimensionais de subgrafo. Posteriormente, definimos um modelo multidimensional em grafo de propriedades que permite produzir modelagens multidimensionais em grafo nos SGBDG.

4.2 MODELO MULTIDIMENSIONAL EM HIPERVÉRTICES

Esse modelo de representação multidimensional utiliza como base o modelo de grafo de propriedade em hipervértices para estruturar os dados dos fatos e dimensões. Nesse

modelo de dados, o hipervértice é um tipo de vértice que pode representar um vértice tradicional ou um vértice com subgrafo, contendo vértices e arestas dentro do hipervértice. Com isso, é possível manipular da mesma forma subgrafos ou vértices tradicionais, utilizando um grafo de hipervértices.

Para facilitar a compreensão, a Figura 26 apresenta um exemplo de hipervértices. Na ilustração, os hipervértices são representados tanto por um retângulo tracejado quanto por um círculo. Eles podem armazenar propriedades e possuir mais de um rótulo. As arestas definem relacionamentos entre os hipervértices podendo, também, armazenar propriedades.

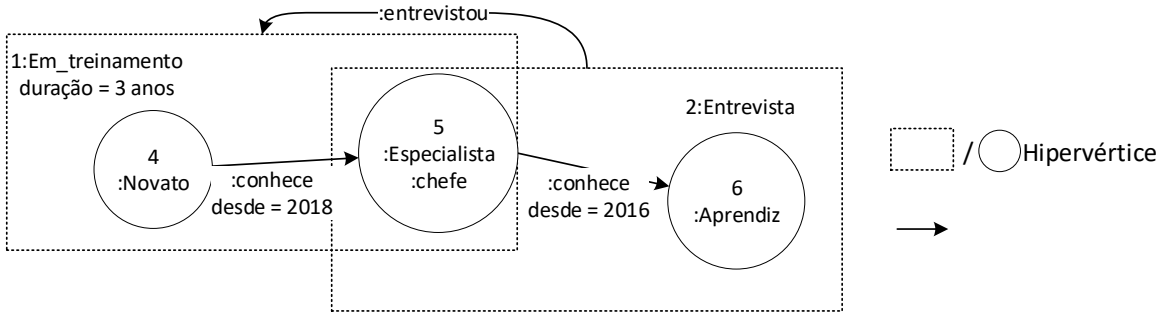


Figura 26 - Representação de dados em hipervértices

Com base na representação do Modelo de Grafo de Propriedade em Hipervértices (MGPH) apresentado no Capítulo 2, a Figura 27 apresenta variações do modelo estrela, tendo como principal mudança os dados que representam os fatos. Neste modelo multidimensional, além da indexação dos fatos em função das dimensões, os fatos correspondem aos dados alvo da análise e podem ser representados no MGPH. Cada fato é representado por um hipervértice que pode se relacionar com outros hipervértices para formar uma rede de informação, o qual denominamos de grafo de fatos.

Definição 1: Grafo de Fatos é a rede de informação criada pelas interconexões entre os conjuntos de dados (fatos) que se deseja analisar. Essa representação serve para qualquer contexto em que os dados alvo da análise possuem relacionamentos entre si. Por exemplo, considerando que os dados alvo da análise (fatos) correspondem aos hipervértices V ∈ MGPH e que os relacionamentos entre esses dados alvo correspondem às arestas A ∈ MGPH; então o MGPH(V,A) corresponde a um Grafo de Fatos, da mesma forma que um MGP(V,A) ao utilizar os vértices V e as arestas A representa, respectivamente, o alvo da análise e seus relacionamento.

Na Figura 27 são ilustradas quatro formas de representar o MMHP. A representação (A) apresenta hipervértices vazios sem relacionamentos entre os hipervértices; a representação (B) apresenta hipervértices sem relacionamentos entre os hipervértices

contendo subgrafos; a representação (C) apresenta hipervértices vazios com relacionamentos entre os hipervértices; e a representação (D) apresenta hipervértices com relacionamentos entre os hipervértices contendo subgrafos.

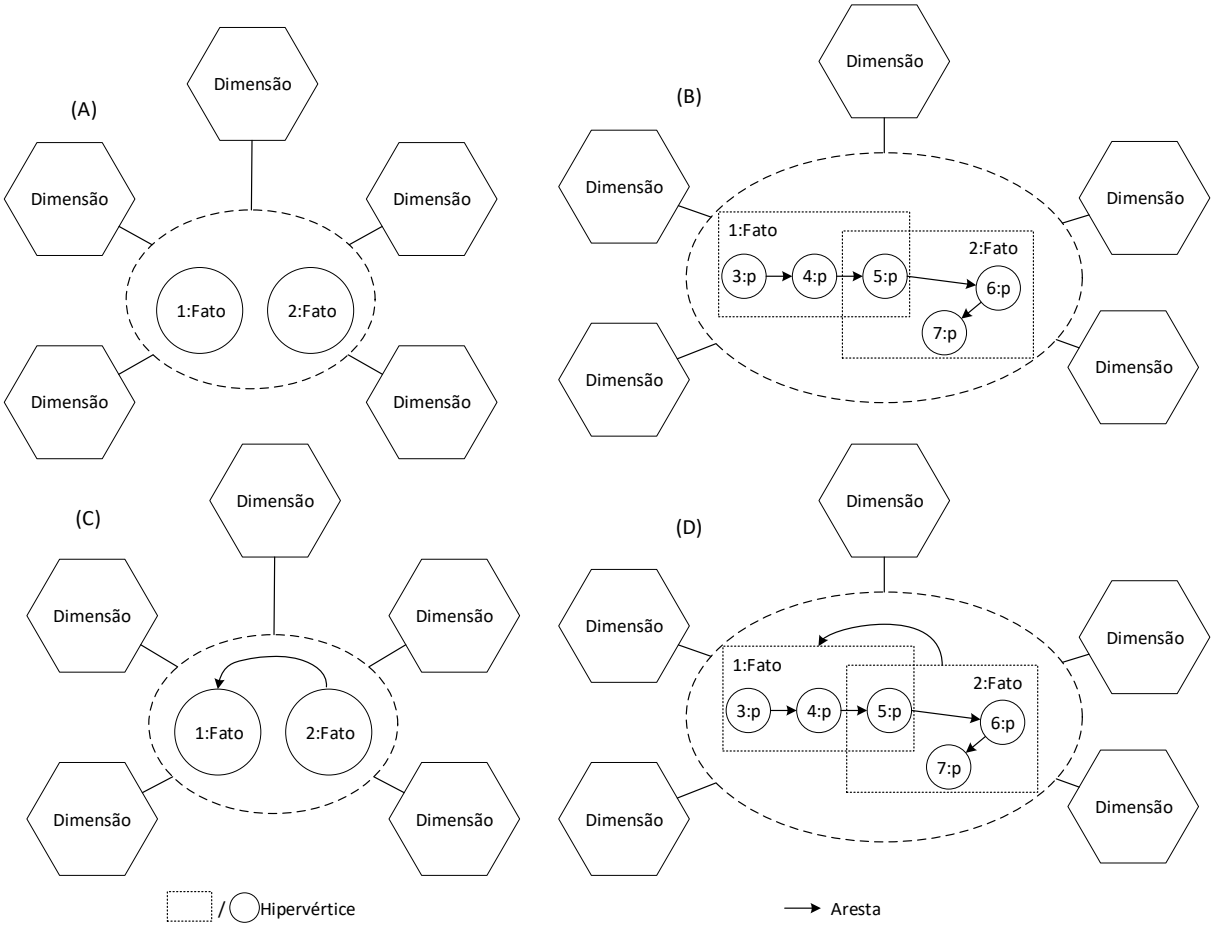


Figura 27 - Modelo Multidimensional em Hipervértices (MMHP)

Nas representações da Figura 27, os círculos e os retângulos pontilhados são hipervértices, as setas são arestas (relacionamentos) entre os hipervértices, os hexágonos são dimensões e a elipse tracejada é o conjunto de fatos composto por hipervértices. Observando essas formas de modelar os dados, percebemos que nas representações A e C os fatos são hipervértices que registram apenas os valores das propriedades e em B e D os fatos são hipervértices com subgrafos que registram tanto as estruturas topológicas de um grafo quanto os valores das propriedades. Além disso, as representações C e D apresentam uma estrutura nova que adiciona relacionamentos entre os fatos de forma a compor um grafo de fatos.

Esse modelo é definido para lidar com os fatos estruturados em grafo, de modo que permite tanto aplicar consultas multidimensionais como, também, algoritmos de análise em grafo, possibilitando combinar essas duas formas de análise.

Apesar das contribuições oferecidas nesse modelo multidimensional em hipervértices, existe um obstáculo que inviabiliza a sua aplicação. Esse obstáculo consiste em encontrar um SGBDG que atenda ao modelo de grafo de propriedade em hipervértices. Devido à complexidade desse modelo, a maioria dos SGBDG não atendem ao MGPH, de modo que inviabiliza a aplicação desse modelo. Com isso, definimos no modelo de grafo de propriedade o Modelo Multidimensional em Padrões de Grafo (Modelo MPGrafo) que estabelece uma estrutura semelhante ao modelo multidimensional em hipervértices. Além disso, o Modelo MPGrafo utiliza uma representação do modelo de grafo de propriedade que é bastante difundido pelos SGBDG (BONIFATI et al., 2018).

4.3 MODELO MULTIDIMENSIONAL EM PADRÕES DE GRAFO

Na composição do Modelo Multidimensional em Padrões de Grafo (Modelo MPGrafo), definimos conceitos para compor uma estrutura semelhante ao modelo multidimensional em hipervértices, no intuito de representar fatos que contenham tanto a estrutura topológica dos subgrafos analisados quanto os valores das propriedades. Na definição do Modelo MPGrafo, é utilizada uma adaptação do Modelo de Grafo de Propriedade (MGP) (BONIFATI et al., 2018). Esse modelo adaptado mantém os principais elementos estruturais do LDBC (Linked Data Benchmark Council) (ANGLES et al., 2017), restringindo as arestas a possuírem apenas um rótulo. Essa restrição diminui a complexidade dos relacionamentos nos dados de entrada, viabilizando a análise topológica. Com isso, o modelo de grafo de propriedade foi definido da seguinte forma.

Definição 2: Modelo de Grafo de Propriedade (MGP) segue a estrutura MGP = (V, A, §n, §h, §m, §u), onde

- $V \subseteq O$ é um conjunto finito de objetos, chamado vértices;
- $A \subseteq O$ é um conjunto finito de objetos, chamado arestas;
- §n : $A \longrightarrow V \times V$ é uma função que atribui a cada aresta um par ordenado de vértices ;
- §h : $V \longrightarrow P(R)$ é uma função que atribui a cada vértice um conjunto finito de rótulos. (a função P(R) representa os conjuntos de rótulos)
- §m : $A \longrightarrow r \mid r \in R$ é uma função que atribui a cada aresta um único rótulo;
- §u : $(V \cup A) \times K \longrightarrow N$ é uma função parcial que atribui valores N para as propriedades chaves K dos objetos (V U A), tal que os conjuntos de objetos V e A são disjuntos (isto é, $V \cap A = \emptyset$;)

Para diferenciar os elementos que compõem o modelo multidimensional em grafo de propriedade dos elementos do grafo de dados, consideramos os seguintes conjuntos na

definição do Modelo MPGrafo:

- **VM** ⊂ **O** um conjunto finito de vértices,
- **AM** ⊂ **O** um conjunto finito de arestas,
- **RM** ⊂ **R** um conjunto finito com todos os rótulos,
- **KM** ⊂ **K** um conjunto de propriedades chave, e
- **NM** ⊂ **N** um conjunto de valores.

Com a adequação ao modelo de grafo de propriedade, a modelagem do Modelo MPGrafo pode ser aplicável nos SGBDG (tais como, Neo4j[12], Apache TinkerPop Blueprints[13], IBM Graph[14] e Sparksee[15]), permitindo o desenvolvimento de modelagens multidimensionais em grafo. A seguir, definimos formalmente as representações dos três principais conceitos do Modelo Multidimensional em Padrões de Grafo (Modelo MPGrafo).

4.3.1 Componente Analítico

O Componente Analítico representa o conceito de fato no Modelo MPGrafo. Ele é uma estrutura composta por um vértice, chamado Vértice Analítico, e um subgrafo, chamado Subgrafo Analítico.

Definição 3: Vértice Analítico (VA) é um vértice VA ∈ VM que é responsável por identificar um fato e armazenar os valores das medidas de análise. Esse vértice é definido por VA = (id, r_{VA}, KA, §p), onde

- id é um identificador único;
- r_{VA} ∈ RM é um rótulo reservado que é comum a todos os vértices VA, sendo essencial no processamento das consultas;
- KA ⊂ KM é um conjunto de propriedades chave que armazena os valores das medidas a serem processadas na consulta; e
- §p: VA x KA ⟶ NM é uma função parcial que atribui valores para as propriedades do vértice.

Definição 4: Meta Grafo (MG) é denotado na forma de MG = {$(Rv_1)^+$ -[Ra]-› Rv_2 ...-›$Rv_{(n+1)}$}, o qual identifica subgrafos por meio de caminhos com fecho transitivo para representar caminhos com ramificações. Nessa denotação, Rv e Ra representam os rótulos dos vértices e arestas, respectivamente, que compõem o caminho do grafo.

[12] http://neo4j.com/
[13] http://tinkerpop.apache.org/
[14] https://ibm-graph-docs.ng.bluemix.net/
[15] http://www.sparsity-technologies.com/

A expressão de fecho transitivo $(Rv_1)^+$ -[Ra]-› Rv_2 especifica a possibilidade de ramificações no caminho, permitindo que um ou mais vértices com rótulo (Rv_1) se relacione com um vértice com rótulo tipo Rv_2. Assim, dizemos que um subgrafo SubG = ({v_1, v_1 ... v_2 ... $v_{(n+1)}$}, {a_{12}, a_{12} ... $a_{n(n+1)}$}) segue o Meta Grafo MG, se ∀i, §h(vi) = Rv_i. Contudo, o Meta Grafo é responsável por identificar subgrafos conexos da base de dados para compor os Componentes Analíticos.

Definição 5: SubGrafo Analítico (SubG) é um recorte do grafo de dados Gd = (V,A), que atende ao Meta Grafo (MG) definido pelo usuário. Esse recorte corresponde a um subgrafo SubG = (SubV,SubA) que especifica a informação a ser analisada. Na representação formal $\forall i, SubG_i \subset Gd \mid SubV_i \subset \mathrm{V} \wedge SubA_i \subset \mathrm{E} \wedge SubG_i \in mg(Gd)$, onde todos os elementos de um subgrafo $subG$ estão contidos no grafo de dados Gd e o conjunto de subgrafos MG(Gd) = {$SubG_1$,..., $SubG_n$} representa todos os subgrafos que atendem ao Meta Grafo (MG).

Definição 6: Componente Analítico (CA) representa uma instância dos fatos que combinam as informações das propriedades e da estrutura topológica do grafo em uma estrutura. Essa estrutura é composta por um Vértice Analítico e um Subgrafo Analítico. Os Vértices Analíticos são utilizados para armazenar as informações que representam o subgrafo e compor o grafo de fatos da modelagem. Os Subgrafos Analíticos são utilizados para representar a estrutura topológica do fato. A formalização dessa estrutura é definida por Ca = (VA, AS, SubG, §am, §ym), onde

- VA ⊂ VM é um conjunto de Vértices Analíticos;
- AS ⊂ AM é um conjunto finito de arestas, denominado relacionamento de subgrafo (relsub), que possui o rótulo reservado r_{relsub} ∈ RM;
- SubG = (SubV,SubA) é um Subgrafo Analítico que representa um recorte da base de dados, de modo que:
    - os conjuntos de objetos SubV e SubA são disjuntos e atendem às especificações do Meta Grafo MG; e
    - o SubG é um grafo conexo, no qual todos os vértices que o compõem são interconectados.
- §ym : AS ⟶ VA x SubV é uma função que atribui a cada aresta o relacionamento entre um Vértice Analítico VA e um vértice SubV que constitui o Subgrafo Analítico SubG; e
- §pm : VA x KM ⟶ NM é uma função parcial que atribui valores para as

propriedades do Vértice Analítico VA. Esses valores representam medidas de análise do Componente Analítico.

Na definição do Componente Analítico, o Vértice Analítico é considerado o principal vértice que identifica os dados alvo da análise (fato). Ele é encarregado de armazenar os valores das medidas de análise, os quais representam as informações do subgrafo, e por formar os relacionamentos analíticos entre os subgrafos. Os valores são armazenados em propriedades para serem processadas nas medidas de análise. Os relacionamentos analíticos relacionam os Vértice Analíticos, de modo a estabelecer na modelagem o grafo de fatos, a qual permite o processamento de medidas topológicas.

Na formação dos Componentes Analíticos é necessário um Meta Grafo MG e um grafo de dados Gd, que atenda ao Modelo de Grafo de Propriedade (Definição 2). O Meta Grafo é encarregado de identificar os subgrafos do grafo de dados para compor o conjunto de subgrafos $MG(Gd) = \{SubG_1, ..., SubG_n\}$. Cada subgrafo do conjunto MG(Gd) forma um Componente Analítico da modelagem, onde $n = |MG(Gd)|$ especifica o número total de subgrafos que constituem os Componentes Analíticos do grafo de dados. Com a identificação de um Subgrafo Analítico SubG para a formação de um Componente Analítico CA é necessário criar um Vértice Analítico VA e os relacionamentos de subgrafos entre o Vértice Analítico criado e todos os vértices que compõem o subgrafo.

4.3.1.1 Pré-processamento

Após a formação estrutural do Componente Analítico, é realizado um pré-processamento no subgrafo para definir valores que podem representar as propriedades e a estrutura topológica do Componente Analítico. Esses valores são armazenados no Vértice Analítico do CA, para o processamento de medidas de análise. Com isso, na formação de cada Componente Analítico é necessário um pré-processamento sobre os dados do Subgrafo Analítico para encontrar valores que possam ser analisados. Dessa forma, consideramos duas formas de pré-processar as informações do subgrafo, as quais denominamos de Pré-processamento de Propriedade (PP) e Pré-processamento Topológico (PT).

O pré-processamento de propriedade utiliza os valores das propriedades contidos nos elementos do subgrafo para representar o subgrafo por inteiro. Nesse pré-processamento, é possível computar valores que servem para analisar as informações do subgrafo. Por exemplo, utilizando um subgrafo para representar a execução de um processo, é possível

comparar os processos somando as propriedades do subgrafo que especificam a duração de tempo.

Já o pré-processamento topológico utiliza algoritmos de análise em grafo para processar os elementos do subgrafo. Os valores encontrados caracterizam os elementos do subgrafo em função de toda a base de dados, proporcionando formas de comparar os subgrafos. Por exemplo, utilizando um subgrafo que representa um grupo de pesquisa, é possível calcular a centralidade de cada integrante do grupo, no intuito de comparar os grupos mais influentes.

4.3.1.2 Exemplo de Componentes Analíticos

Para exemplificar a formação de um Componente Analítico, utilizamos os dados da amostra do DBLP para compor dois exemplos. A Figura 28 apresenta o exemplo de um Componente Analítico que atende ao Meta Grafo MG1 = { Rv | λ(Rv) = :Person}. Na imagem, aparece a amostra do DBLP com os Subgrafos Analíticos que foram identificados pelo MG1. Nesse exemplo, foi selecionado o subgrafo com o vértice 102 para constituir um Componente Analítico. Na criação desse CA é realizado o pré-processamento no subgrafo, a fim de definir valores e propriedades para as medidas de análise. No exemplo, o Vértice Analítico possui as seguintes propriedades: ‘salary’ com valor 17000, que foi obtido do subgrafo sem o pré-processamento; e ‘degree’ com valor 0,6, que realizou o Pré-processamento Topológico (PT) para calcular o grau de centralidade do vértice 102.

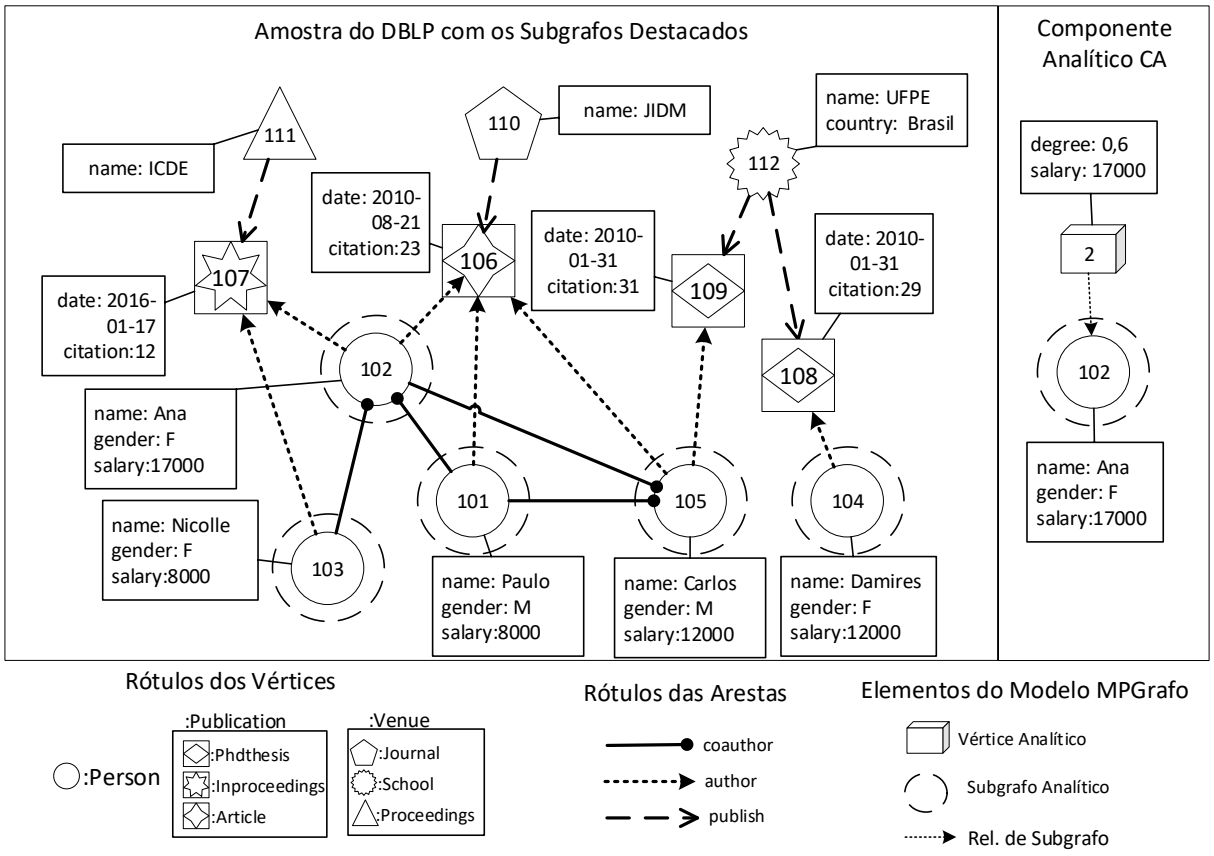


Figura 28 - Componente Analítico com um vértice no Subgrafo Analítico

A Figura 29 apresenta o exemplo de um componente analítico que atende ao Meta Grafo MG2 = { $(Rv_1)^+$ -[Ra_{12}]-› Rv_2 ‹-[Ra_{23}]- Rv_3 | $\lambda(Rv_1)$ = :Person Λ $\lambda(Rv_2)$ = :Publication Λ $\lambda(Rv_3)$ = :Venue Λ $\lambda(Ra_{12})$ = :author Λ $\lambda(Ra_{23})$ = :publish }. Na imagem, aparece a amostra do DBLP com os Subgrafos Analíticos que foram identificados pelo MG2. Nesse exemplo, foi selecionado o subgrafo com os vértices {101,102,105,106,110} para constituir um Componente Analítico. Na criação desse CA é realizado o pré-processamento no subgrafo, a fim de definir valores e propriedades para as medidas de análise. No exemplo, o Vértice Analítico possui as seguintes propriedades: ‘sumSalary’ com valor ‘37000’, que realizou um pré-processamento de propriedade (PP) para calcular a soma de todos os salários das pessoas do subgrafo; ‘sumDegree’ com valor ‘1,75’, que realizou um PT para calcular a soma do grau de centralidade dos vértices com rótulo (:Person), ‘author’ com valor ‘3’, que foi obtido pelo número de vértices com rótulo (:Person) no subgrafo e ‘citation’ com valor 23, que foi obtido do vértice 106 do subgrafo sem pré-processar.

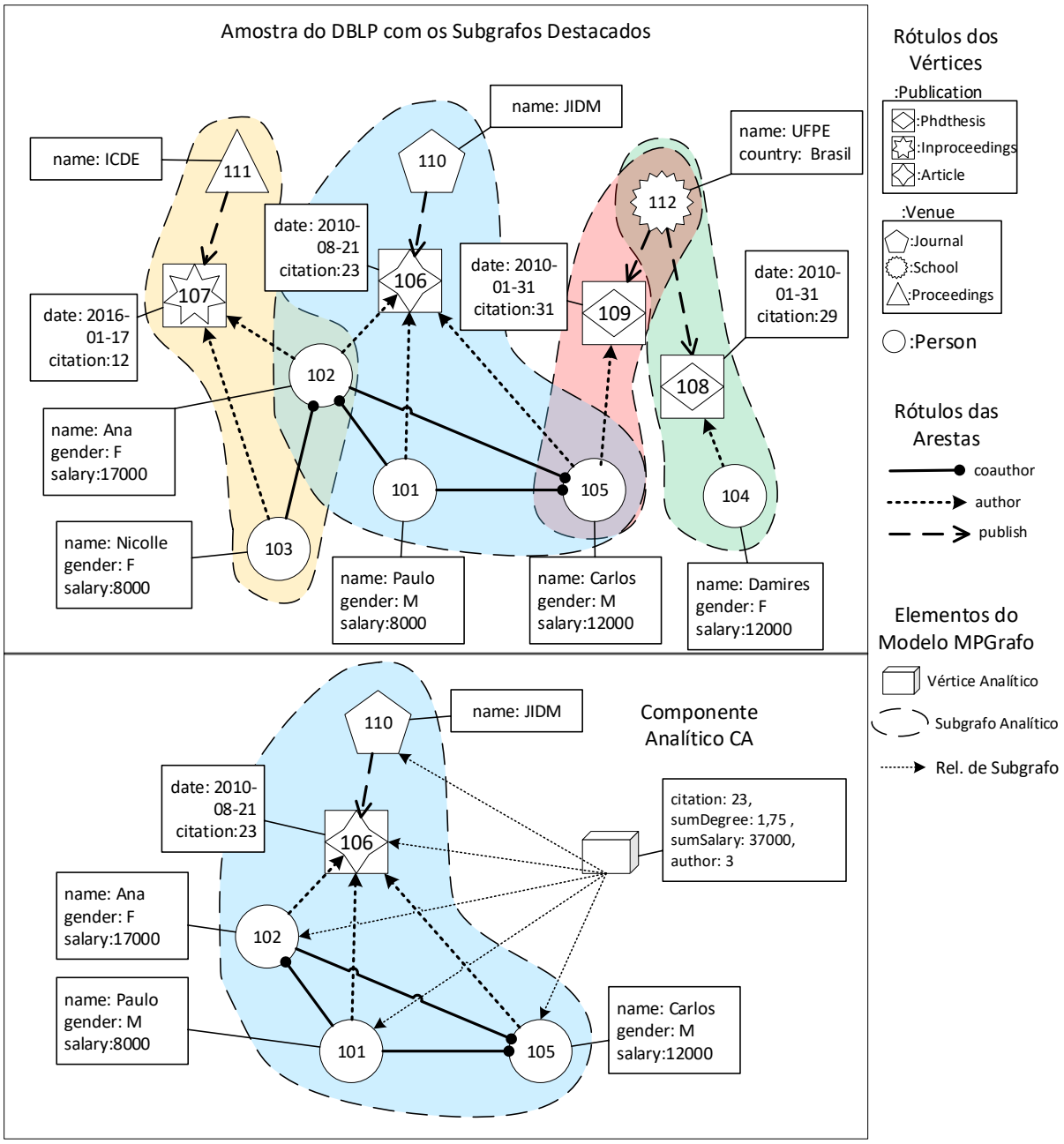


Figura 29 - Componente Analítico com um conjunto de vértices no Subgrafo Analítico

4.3.2 Medidas de Análise

As medidas de análise são algoritmos que extraem o conhecimento dos fatos nas consultas multidimensionais, considerando os valores das propriedades e os relacionamentos entre os Vértices Analíticos no processamento das medidas. Na literatura, os trabalhos sobre Graph OLAP (CHEN et al., 2008, 2009) definiram diferentes classificações para os tipos de medidas de análise, tais como, *distributive*, *algebraic* e *holistic* em (CHEN et al., 2009) e *Content-Based* e *Graph-Specific* em (GHRAB et al., 2015). Na AAMPGrafo, definimos três tipos de medidas de análise que foram categorizadas em

função dos tipos de dados computados. Apresentamos a seguir os tipos de medidas definidas.

Definição 7: Medida de Propriedade (MP) utiliza os valores que caracterizam os subgrafos dos Componentes Analíticos. Este tipo de medida é similar às medidas do DW tradicional, pois não utiliza informações topológicas do grafo no seu processamento. Por exemplo, se um Componente Analítico (fato) representar uma avaliação de um filme, é possível definir uma medida para calcular a média das avaliações.

Definição 8: Medida Topológica (MT) utiliza a topologia do grafo para processar valores numéricos que representem a estrutura topológica do grafo. Por exemplo, há algoritmos para calcular medidas do grafo (tais como, Eigenvector Centrality, *Degree Centrality*, *Betweenness Centrality*, *Closeness Centrality*) (BORGATTI e EVERETT, 2006; FREEMAN, 1978; NEWMAN, 2010; OPSAHL et al., 2010). Essas medidas citadas utilizam os vértices e as arestas para calcular a centralidade dos vértices. Além disso, existem algoritmos de análise em grafo que avaliam a importância ou similaridade dos elementos em função da estrutura do grafo, tais como, *PageRank* e *SimRank* (PAGE et al., 1999)(JEH e WIDOM, 2002). Esse tipo de medida só pode ser aplicado em uma modelagem que atenda ao modelo MPGrafo e contenha um grafo de fatos para possibilitar analises na estrutura topológica do grafo.

Definição 9: Medida Híbrida (MH) é um tipo de medida que utiliza, no seu processamento, a topologia do grafo de fatos e os valores que caracterizam os subgrafos dos CA. Por exemplo, uma medida que define a influência de um autor, considerando o coeficiente das publicações e os valores de centralidades do grafo de coautores.

Considerando as medidas de análise e a formação dos Componentes Analíticos (fatos), constatamos a necessidade de pré-processar os valores das medidas de análise na construção da modelagem. Esse pré-processamento define valores que podem representar a estrutura topológica e os valores de propriedade encontrados no subgrafo do CA. Para isso, categorizamos as formas de pré-processar na seção do Componente Analítico.

4.3.3 Dimensão

No Modelo MPGrafo, a Dimensão é um conjunto de vértices que contêm informações específicas de uma perspectiva de análise. Essas informações indicam características comuns que distinguem e identificam os Componentes Analíticos (fatos) em função das

suas especificidades. Com isso, esse componente funciona como um indexador para categorizar os Componentes Analíticos (fatos) em função das suas características em uma especifica perspectiva. Portanto, cada Dimensão destaca uma perspectiva de análise e a combinação de diferentes Dimensões identifica os Componentes Analíticos por meio de múltiplas perspectivas de análise.

Definição 10: Vértice Dimensional (VD) é um vértice que contêm propriedades que categorizam e especificam as informações dos Componentes Analíticos (fatos). O vértice dimensional é definido por VD = (r_{VD}, KD, §p), onde

- r_{VD} ∈ RM é o rótulo que determina a Dimensão desse VD;
- KD ⊂ KM é um conjunto de propriedades chave que especificam as características da Dimensão; e
- §p : VD x KD ⟶ NM é uma função parcial que atribui valores para as propriedades dos vértices que compõem a Dimensão.

Definição 11: Dimensão (D) é um conjunto de vértices que contêm propriedades que categorizam e especificam as informações dos Componentes Analíticos (fatos). A dimensão é definida por D = (VD, r_{VD},§r), onde

- VD ⊂ VM é um conjunto de Vértices Dimensionais (VD) que representam instâncias de uma Dimensão.
- r_{VD} ∈ RM é um rótulo reservado que é comum a todos os vértices VD de uma dimensão;
- §r : VD ⟶ r_{VD} um função que atribui o mesmo rótulo para todos os Vértices Analíticos de uma mesma Dimensão.

Uma perspectiva é representada por uma Dimensão, contendo um conjunto de vértices com determinadas propriedades e rótulos. Cada informação especificada de uma perspectiva é representada por um vértice que contém valores únicos da perspectiva. Por exemplo, na perspectiva de tempo a data 2016-01-17 é o valor de menor granularidade que é responsável por identificar os fatos ocorridos nessa data. Com isso, é definido um vértice v ∈ VD para registrar a menor granularidade dessa perspectiva e vincular todos os Componentes Analíticos que atendem a essa informação. Além disso, dependendo da perspectiva tratada na Dimensão é possível que o vértice v contenha propriedades categóricas, com menor grau de detalhe, para permitir agregações com maior granularidade.

Para facilitar a compreensão da Dimensão, apresentamos na Figura 30 um exemplo

utilizando as informações de data = {2016-01-17, 2010-01-31}. A imagem mostra a Dimensão contendo dois Vértices Dimensionais VD para registrar as informações de data e estabelecer um relacionamento com os Componentes Analíticos, denominado relacionamento dimensional.

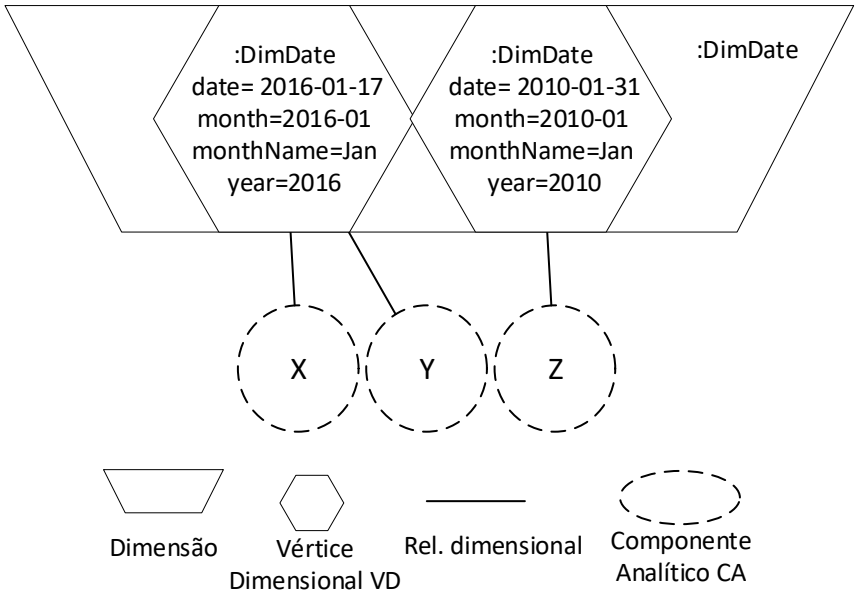


Figura 30 - Representação estrutural da Dimensão

Na Figura 30, cada vértice é constituído por quatro propriedades, uma propriedade registra a informação de menor granularidade, como a propriedade “date”, e as outras propriedades podem especificar as informações mais genéricas, tais como: “month”, “monthName” e “year”. Essa forma de modelar os dados no grafo permite identificar os Componentes Analíticos que se relacionam com os Vértices Dimensionais cujos valores de propriedade são requisitados nas consultas. Por exemplo, quais os Componentes Analíticos que ocorreram na data 2016-01-17? Essa consulta retorna os CA = {X e Y}. Quais Componentes Analíticos ocorreram em janeiro? Retorna CA = { X, Y e Z}.

4.3.4 Especificação do Modelo MPGrafo

Após as definições dos componentes que constituem o Modelo MPGrafo, apresentaremos a seguir a definição formal do mesmo.

Definição 12: Modelo Multidimensional em Padrões de Grafo (Modelo MPGrafo) é definido por (D, AD, CA, AA, §dm, §am, §pm), onde

- D é um conjunto finito de Dimensões;
- AD ⊂ EM é um conjunto finito de arestas, denominado “relacionamento dimensional” (reldi), que vincula os Vértices Dimensionais com os

Componentes Analíticos.

- CA é um conjunto finito de Componentes Analíticos;
- AA ⊂ EM é um conjunto finito de arestas, denominada “relacionamento analítico” (relan), que representa os relacionamentos entre os Subgrafos Analíticos da base de dados. Além disso, relacionam os Componentes Analíticos por meio dos relan entre os Vértices Analíticos.
- §dm : AD ⟶ D (VD) x CA(VA) é uma função que atribui a cada aresta uma relação entre um Vértice Dimensional VD e um Vértice Analítico VA;
- §am : AA ⟶ CA(VA) x CA(VA) é uma função que atribui a cada aresta um par ordenado de Vértices Analíticos para representar o relacionamento entre Componentes Analíticos.
- §pm : AA x KM ⟶ NM é uma função total que atribui a aresta AA as mesmas propriedades e valores que os relacionamentos representados na base de dados.

Após apresentar todas as definições do Modelo Multidimensional do Grafo de Propriedade, introduziremos as formas de modelar as fontes de dados de grafo de propriedades usando o Modelo MPGrafo.

4.3.5 Formas de usar o Modelo MPGrafo

No intuito de mostrar as diferentes formas de utilizar o Modelo MPGrafo, apresentamos na Figura 31 quatro imagens que destacam os principais componentes de uma modelagem e as diferentes formas de usar o Modelo Multidimensional em Grafo de Propriedade. Nessa figura, as quatro imagens se assemelham às imagens do Modelo Multidimensional em Hipervértices (MMHP) apresentadas na Figura 27, na qual cada imagem possibilita formas de analisar os dados de um grafo de dados. A imagem (A) consiste em uma modelagem que não contém relacionamentos entre os Componentes Analíticos, os quais são constituídos por um subgrafo com apenas um vértice. A imagem (B) consiste em uma modelagem que não contém relacionamentos entre os Componentes Analíticos, os quais são constituídos por um subgrafo com vários vértices. A imagem (C) consiste em uma modelagem que possui relacionamentos entre os Componentes Analíticos, os quais são constituídos por um subgrafo com apenas um vértice. A imagem (D) consiste em uma modelagem que possui relacionamentos entre os Componentes Analíticos, os quais são constituídos por um subgrafo com vários vértices.

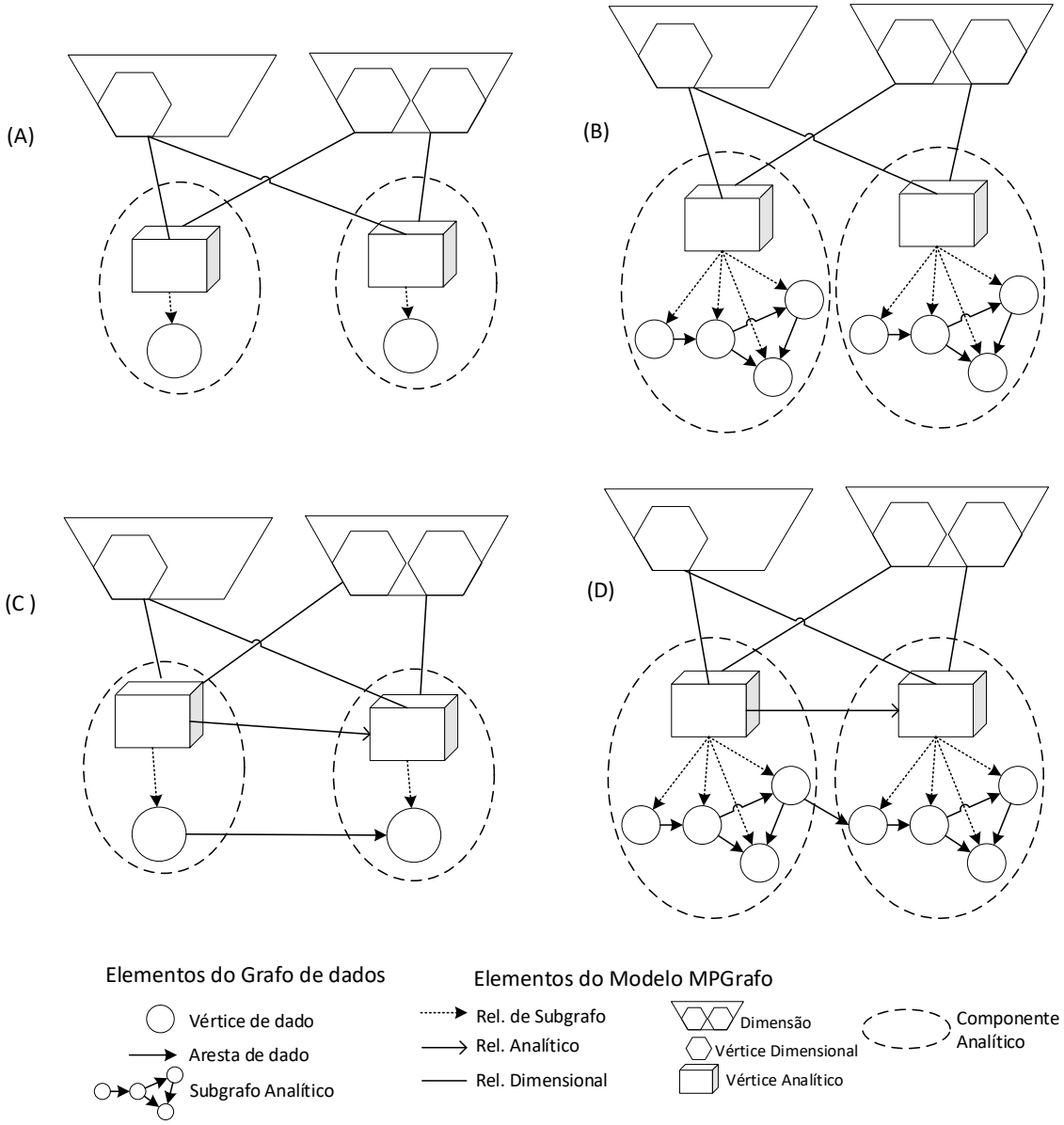


Figura 31 - Exemplo de Modelagens Multidimensionais em Padrões de Grafo

Essas formas de modelagem são semelhantes às apresentadas na MMHP, além de serem aplicáveis no SGBDG. Cada forma de usar o Modelo MPGrafo habilita determinados tipos de medidas de análise, de modo que: a imagem (A) habilita medidas do tipo MP (Medida de Propriedade) e permite pré-processamento do tipo PP; a imagem (B) habilita medidas do tipo MP e permite pré-processamento do tipo PP e PT; a imagem (C) habilita medidas do tipo MP, MT (Medida Topológica) e MH (Medida Híbrida) e permite pré-processamento do tipo PP; e a imagem (D) habilita medidas do tipo MP, MT e MH e permite pré-processamento do tipo PP e PT.

A aplicação desses formatos contribui para a realização de consultas multidimensionais e

análises em grafo por meio do SGBDG. Nas imagens (A) e (B) da Figura 31 é possível analisar os Componentes Analíticos utilizando as Dimensões para especificar as múltiplas perspectivas nas consultas. Nas imagens (C) e (D), além de possibilitar o uso de múltiplas perspectivas com as Dimensões, é possível utilizar os relacionamentos entre os Componentes Analíticos para processar algoritmos de análise em grafo, permitindo associar consultas de múltiplas perspectivas com a análise em grafo.

4.3.6 Aplicação do Modelo MPGrafo

No intuito de entender as etapas de produção da modelagem no Modelo MPGrafo, apresentamos na Figura 32 um diagrama de atividade UML para especificar a sequência de atividades necessárias para o desenvolvimento dessa modelagem. Em outras palavras, o diagrama descreve a sequência de atividades que um desenvolvedor precisa seguir para construir no SGBDG uma modelagem que atenda ao Modelo MPGrafo.

A produção da modelagem no Modelo MPGrafo consiste em inserir elementos estruturais, que constituem as Dimensões e os Componentes Analíticos, no grafo de dados. A inserção desses elementos não altera o conteúdo original do grafo de dados, ao mesmo tempo que modela os dados. Em decorrência disso, consideramos o Modelo MPGrafo adaptativo a novos dados, visto que a modelagem multidimensional é produzida por meio de inserções e vinculações dos elementos estruturais aos dados inseridos. Além disso, a característica adaptativa do modelo também permite modificar ou remover a estrutura sem alterar o conteúdo dos dados.

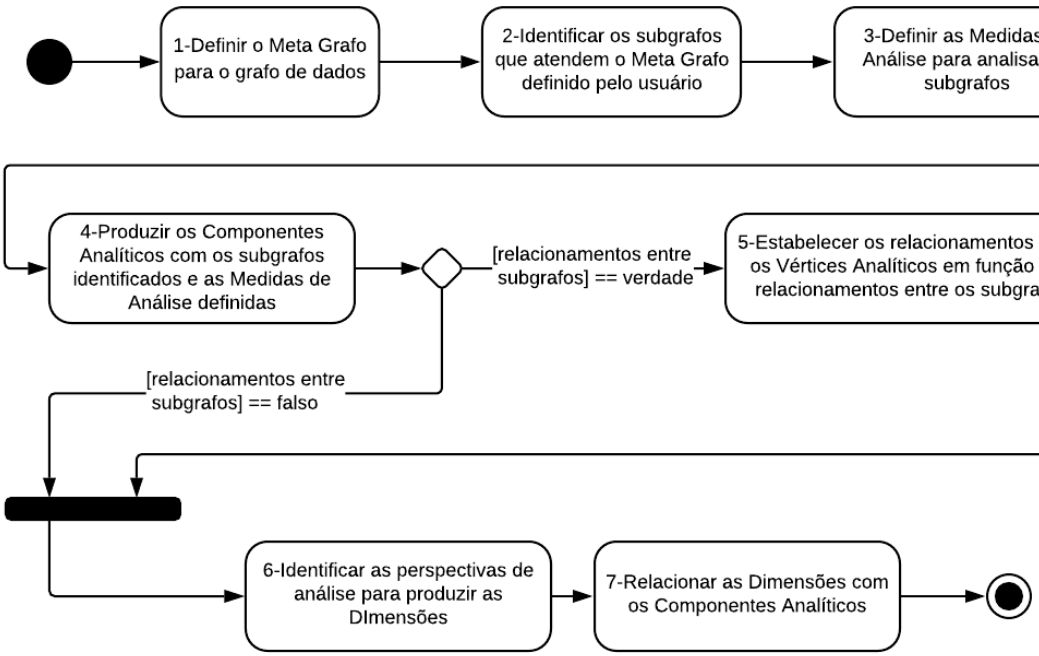


Figura 32 - Diagrama de Atividade para produção de uma Modelagem MPGrafo

Para facilitar o entendimento desse processo de modelagem, apresentamos na Figura 33 uma sequência de imagens que retratam as mudanças do grafo de dados durante a construção da modelagem. Cada imagem contém um conjunto de números entre chaves que especifica as atividades realizadas no diagrama de atividade da Figura 32. Essas numerações informam quais atividades foram realizadas para constituir a modelagem ilustrada. No processo de modelagem apresentado na Figura 33, são utilizados um grafo de dados e dois Meta Grafos para mostrar a sequência de transformações de uma modelagem ilustrando quatro formas diferentes de modelar um grafo de dados com o Modelo MPGraph. Nesse processo, as imagens finais com as letras (A, B, C, D) são representações detalhadas da modelagem das imagens da Figura 31 com as mesmas letras. Na descrição da Figura 33, a numeração nos Vértices Analíticos são identificadores únicos. Os trapézios invertidos representam as Dimensões e cada hexágono contido representa um Vértice Dimensional VD, que especifica as características de uma dimensão.

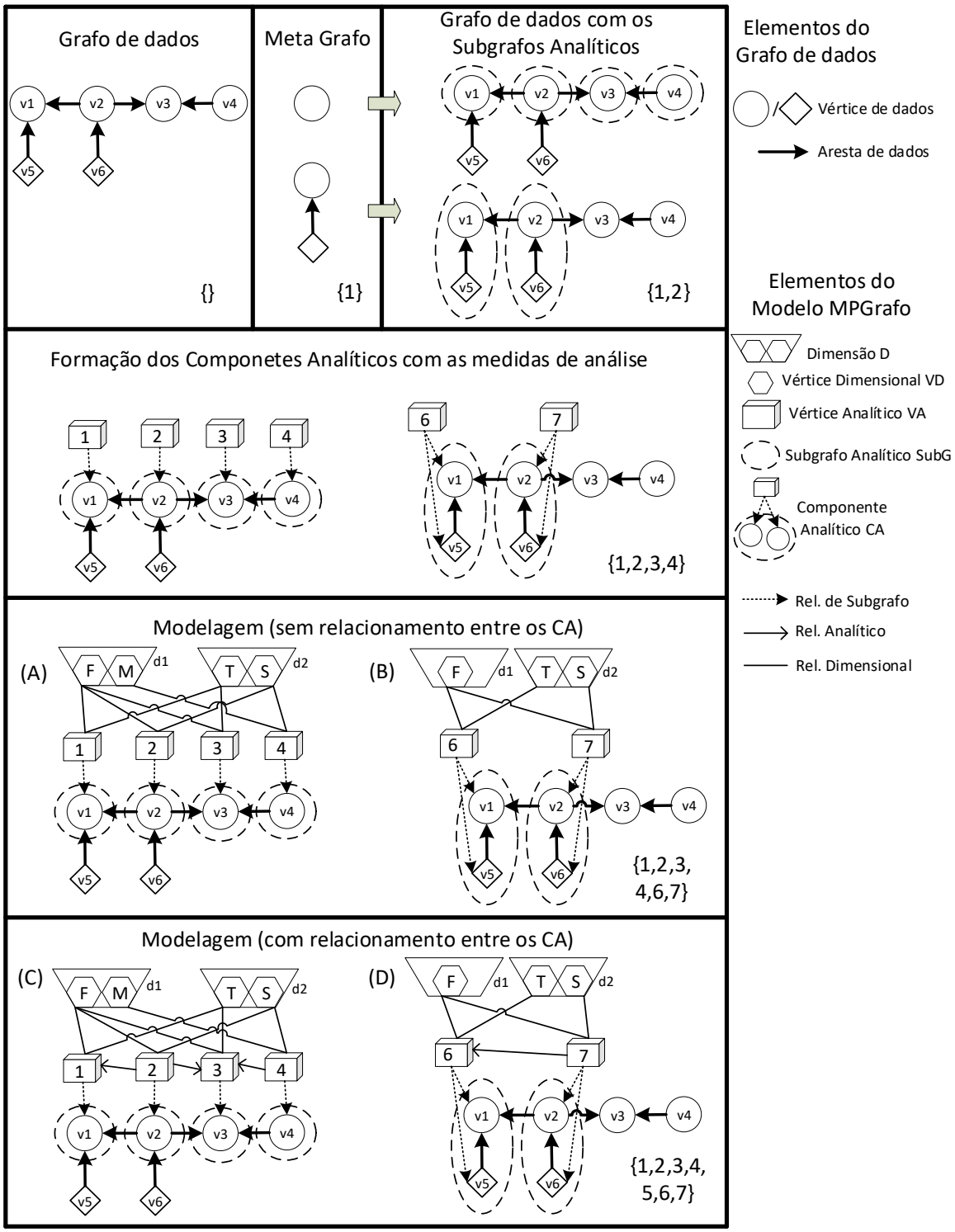


Figura 33 - Processo de modelagem no Modelo MPGrafo

4.4 DISCUSSÃO DO MODELO MPGRAFO

Com base nessas especificações da modelagem, consideramos que a AAMPGrafo necessita manter o grafo em conformidade com o Modelo MPGrafo para possibilitar as consultas de análise. Apesar dos elementos estruturais do Modelo MPGrafo não alterarem

o conteúdo do grafo de dados, a AAMPGrafo necessita de uma integração entre os elementos estruturais e os dados a serem analisados. Com isso, definimos dois cenários que podem ser utilizados para estruturar os dados na AAMPGrafo.

O primeiro consiste em estruturar completamente um grafo de dados proveniente de uma rede de informação. Nesse cenário, é necessário modelar o grafo de dados por inteiro para uma modelagem que obedece ao Modelo MPGrafo.

O segundo corresponde a manter a estrutura dos dados em conformidade com o Modelo MPGrafo, considerando o incremento dos dados. Nesse cenário, é realizado um tratamento, parecido ao ETL do Data Warehouse (KIMBALL e ROSS, 2002a), para deixar os dados inseridos em conformidade com a modelagem já definida. Nesse tratamento, os dados inseridos são analisados e relacionados aos elementos estruturais correspondentes, de modo que a modelagem dos dados mantenha a conformidade com o Modelo MPGrafo. Esses cenários apresentam situações de tratamento de dados semelhantes as encontradas no Data Warehouse. Com isso, consideramos que a manutenção da modelagem restringe um pouco a flexibilidade do grafo, mas proporciona a realização de análise multidimensionais em grafos.

4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, mostramos uma visão geral da AAMPGrafo com uma introdução breve do seu funcionamento. Apresentamos a amostra de dados do repositório DBLP, que utilizamos na explanação dessa abordagem. Definimos o modelo multidimensional em padrões de grafo (MPGrafo), detalhando cada conceito que o compõe. Especificamos formas de modelar o grafo de dados usando o Modelo MPGrafo. Por fim, discutimos sobre a aplicação e manutenção do Modelo MPGrafo em um grafo de dados. No próximo capítulo apresentamos os tipos de consulta e as etapas necessárias na realização de consultas multidimensionais em grafo, mostrando um modelo de consulta e de representação gráfica, que definimos para ser usado no SGBDG.

5 CONSULTA MULTIDIMENSIONAL EM PADRÕES DE GRAFO

A combinação de diferentes recursos de análise na mesma consulta está sujeita a um processamento com alta interação entre diferentes algoritmos de análise. Na AAMPGrafo, definimos o modelo de Consulta Multidimensional em Padrões de Grafo (CMPGrafo) para receber os parâmetros de configuração de uma consulta e definir uma sequência de etapas para o processamento dessa consulta. Neste trabalho, não definimos uma linguagem de consulta multidimensional em grafo para integrar operações ou especificações de análise, podendo ser o escopo de um trabalho futuro. Dessa forma, definimos uma função parametrizada que recebe a configuração de consulta para executar uma consulta multidimensional em grafo utilizando os recursos do SGBDG.

Este capítulo cobre o seguinte conteúdo. A Seção 5.1 apresenta os parâmetros de configuração utilizados nas consultas. A Seção 5.2 apresenta uma visão geral do modelo de CMPGrafo. A Seção 5.3 apresenta a etapa de Seleção, mostrando como utilizar os recursos de recuperação de informação do SGBDG. A Seção 5.4 apresenta a etapa de Análise Topológica, mostrando os algoritmos de análise em grafo. A Seção 5.5 apresenta a etapa de Agregação e Visualização, mostrando o processo de agregação de dados em grafo. A Seção 5.6 apresenta exemplos de consulta na AAMPGrafo. Por fim, a Seção 5.7 conclui o capítulo com as considerações finais.

5.1 PARÂMETROS DE CONFIGURAÇÃO DE CONSULTA

Na configuração de consulta, definimos 4 tipos de parâmetros: Consulta do SGBDG, Configuração dimensional, Configuração topológica e Configuração de medida.

- Consulta do SGBDG: é um parâmetro que recebe uma consulta com a linguagem nativa do SGBDG. Essa consulta precisa seguir um modelo de consulta que atenda ao Modelo MPGrafo. Esse parâmetro é responsável por recuperar os dados do SGBDG que serão analisados na consulta.
- Configuração dimensional: é um parâmetro que específica quais rótulos e propriedades serão utilizados na unificação dos Vértices Dimensionais durante a consulta.
- Configuração topológica: é um parâmetro que específica quais algoritmos de análise em grafo serão processados sobre o grafo de fatos constituído pelos Vértices Analíticos recuperados. Além disso, permite utilizar funções de agregação sobre os

resultados desses algoritmos.

- Configuração de medida: é um parâmetro que específica quais funções de agregação (count, avg, sum, min, max) serão aplicadas nas propriedades e relacionamentos dos vértices que foram recuperados.

5.2 VISÃO GERAL

A aplicação do Modelo MPGrafo na AAMPGrafo possibilita usar os recursos do SGBDG no processamento de consulta, definindo o modelo CMPGrafo. Esse modelo é encarregado de processar a consulta utilizando três etapas de processamento: Seleção, Análise Topológica, e Agregação e Visualização.

A etapa de Seleção utiliza os recursos do SGBDG para realizar consultas em grafo, no intuito de recuperar os dados para a análise. Nessa etapa, as consultas submetidas seguem um modelo de consulta que permite identificar e recuperar os Componentes Analíticos (fatos) em função das expressões definidas na consulta. Além disso, essa etapa utiliza o parâmetro de configuração dimensional para organizar os vértices dimensionais que compõem a consulta. Ao término dessa etapa, os Componentes Analíticos e as Dimensões são recuperados e disponibilizados para as próximas etapas de processamento.

A etapa de Análise Topológica necessita de uma modelagem que atenda ao Modelo MPGrafo e contenha um grafo de fatos para ser possível realizar análises topológicas. Além disso, a execução dessa etapa é optativa, pois requer a solicitação de análise topológica nos parâmetros da consulta. O processamento dessa etapa consiste em executar algoritmos de análise em grafo sobre o grafo de fatos, que é composto pelos Vértices Analíticos recuperados na etapa de Seleção. Após o processamento dessa etapa, os resultados da análise são inseridos nos Vértices Analíticos como propriedades.

A etapa de Agregação e Visualização é responsável por agregar os dados processados nas etapas anteriores e montar uma representação visual da consulta. A agregação dos dados consiste em agregar os Componentes Analíticos que possuem as mesmas características indicadas nas dimensões. A agregação dos Componentes Analíticos consiste de dois algoritmos: Clusterização de Componentes Analíticos (ClusCA) e Agregação dos Componentes Analíticos (AgrCA). O algoritmo ClusCA é encarregado de agrupar os CA que possuem as mesmas características nas dimensões. O algoritmo AgrCA é encarregado de processar as medidas de análise requisitadas na consulta e agregar os

Vértices Analíticos e os Subgrafos Analíticos para compor a resposta da consulta. Durante a execução desses algoritmos, uma representação gráfica é criada para representar a resposta da consulta.

Para melhorar o entendimento, apresentamos na Figura 34, as etapas de processamento de uma consulta utilizando dois exemplos de modelagem. O exemplo A apresenta uma modelagem com um único vértice no subgrafo. Nesse exemplo, a consulta recupera os Componentes Analíticos que se relacionam com os Vértices Dimensionais de duas dimensões diferentes F ou S. O exemplo B apresenta uma modelagem com dois vértices no Subgrafo Analítico. Nesse exemplo, a consulta recupera os Componentes Analíticos que se relacionam com o Vértice Dimensional F.

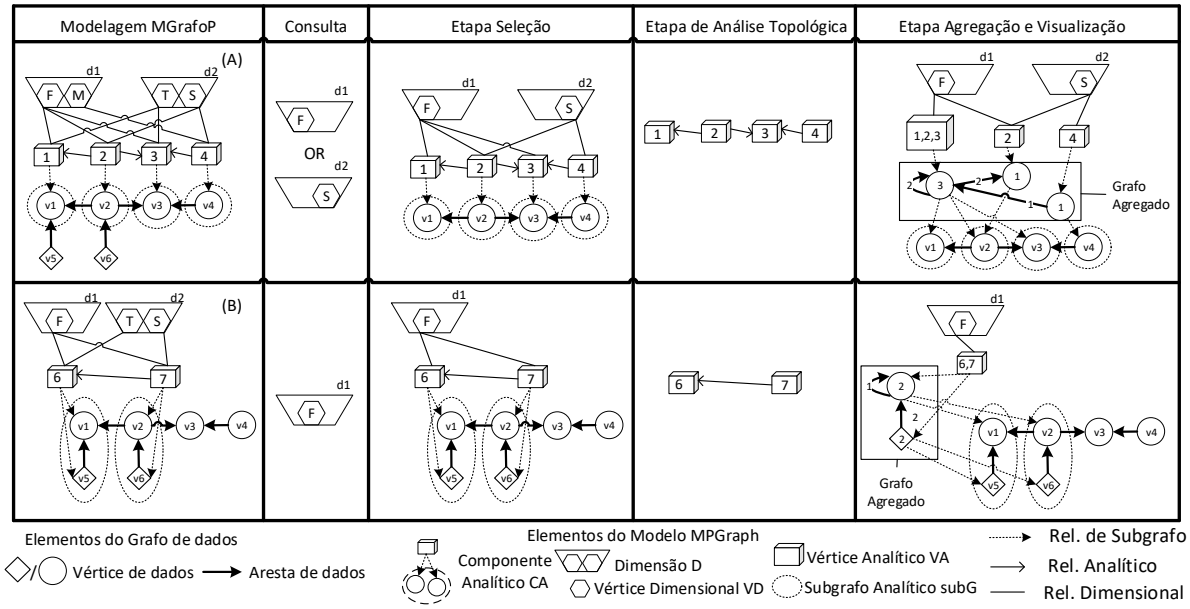


Figura 34 - Processamento do Modelo de CMPGrafo

Na etapa de Análise Topológica, os algoritmos de análise em grafo são executados sobre o grafo de fatos formado pelos Vértices Analíticos que foram recuperados na etapa de Seleção. Na etapa de Agregação e Visualização, é realizado o processo de agregação sobre os Componentes Analíticos recuperados. Na Figura 34, cada Vértice Analítico Agregado apresenta os identificadores dos Vértices Analíticos que o compõem e os Subgrafos Agregados apresentam em cada elemento agregado pesos para indicar a quantidade de elementos que o compõem. No final desse processo, os resultados das consultas mostram uma sequência de relacionamentos que relaciona os Vértices Dimensionais com os Vértices Analíticos Agregados, os Vértices Analíticos Agregados com os Subgrafos Agregados e os vértices dos Subgrafos Agregados com os vértices do grafo de dados.

Portanto, introduzimos na Figura 34 uma visão geral do processo de consulta para melhorar a compreensão das diferentes etapas de processamento da consulta. Na continuação do trabalho, detalharemos cada etapa de processamento seguindo a sequência de execução da CMPGrafo.

5.3 ETAPA DE SELEÇÃO

Na etapa de Seleção, introduzimos formas de selecionar os dados no Modelo MPGrafo utilizando os recursos do SGBDG. Definimos um modelo de consulta para selecionar e recuperar os dados em qualquer SGBDG e mostramos o uso do parâmetro de Configuração Dimensional na consulta.

A seleção dos dados no Modelo MPGrafo consiste em recuperar uma sequência de relacionamentos que interliga os Vértices Dimensionais com os Vértices Analíticos e os Vértices Analíticos com os vértices do grafo de dados, os quais são representados por um Subgrafo Analítico. Com isso, o Vértice Analítico é o principal elemento na consulta do Modelo MPGrafo, visto que possui relacionamentos com os Vértices Dimensionais e os Subgrafos Analíticos. Assim, definimos três formas de selecionar os Vértices Analíticos nas consultas.

- Seleção por dimensão: consiste em identificar os Vértices Dimensionais VD da dimensão para encontrar os Vértices Analíticos VA que são relacionados por meio de uma aresta. Essa forma de seleção identifica os VD por meio de expressões que utilizam rótulo e propriedades das Dimensões. As expressões definidas podem retornar vários VD. Por exemplo, retornar os VD com o rótulo “DimDate” que possuem a propriedade “year” com valor ‘2010’.
- Seleção por propriedade: consiste em identificar os Vértices Analíticos utilizando uma expressão sobre as propriedades do Vértice Analítico. Por exemplo, retornar todos os vértices analíticos que possuem a propriedade ‘salary’ com valor maior que 10000;
- Seleção por relacionamento: consiste em especificar expressões considerando os relacionamentos analíticos entre Vértices Analíticos VA e os relacionamentos dimensionais entre VD e VA. Essa forma de seleção identifica os VA por meio de uma expressão que utilize os relacionamentos como parâmetro. Por exemplo, retornar todos os vértices analíticos que possuem mais de ‘1’ relacionamento com a dimensão “DimPublicationType”.

Com base nas formas de seleção, definimos um modelo de consulta para padronizar as consultas do parâmetro “Consulta do SGBDG”. Esse modelo de consulta foi definido utilizando as especificações da álgebra regular de grafo de propriedade “*Regular Property Graph Algebra”* (RPGA) (BONIFATI et al., 2018).

A definição da RPGA consiste de expressões compostas por elementos de **R** (conjunto de todos os rótulos do grafo), usando operações de fecho transitivo, de união e de junção em grafo. Além disso, a RPGA utiliza uma gramática para representar o conjunto de subconsultas (subRPGA) que compõem as expressões. A RPGA segue a seguinte especificação:

$e := r \mid e^* \mid e \cup e \mid \bowtie^{\phi,\ c}_{pos_i,\ pos_j} (e, \dots, e)$, onde

- $r \in$ R;
- $(e, \dots, e)$ de comprimento n > 0;
- $c \in C$ é um identificador de contexto;
- $pos_i, pos_j \in \{src_1, trg_1, \dots, src_n, trg_n\}$; e
- Φ é uma conjunção de um número finito de termos do formato:
  - $\lambda(pos) = r \mid pos \in \{src_1, trg_1, \dots, src_n, trg_n\}\ and\ r \in$ R;
  - $pos_i.p\ \theta\ pos_j.q\ \ or\ pos_i.p\ \theta\ val \mid\ pos_i,\ pos_j \in \{src_1, trg_1, \dots, src_n, trg_n, edge_1 \dots, edge_n\},\ \{p, q\} \in K\ ,\ val \in N\ and\ \theta \in \{=, \neq, <, >, \leq, \geq\},\ or$
  - $pos_i = pos_j \mid pos_i, pos_j \in \{src_1, trg_1, \dots, src_n, trg_n\}$.

As expressões e ∈ RPGA seguem o formato $\bowtie^{\phi}_{\overline{pos}}(e_1, \dots, e_n)$, onde n > 0, cada e_i é uma expressão de subRPGA (1 ≤ i ≤ n), ϕ é uma conjunção de termos, $\overline{pos}$ é uma lista de comprimento zero ou mais, contendo os elementos de $\{src_1, trg_1, \dots, src_n, trg_n\}$.

5.3.1 Modelo de Consulta em Grafo para o Modelo MPGrafo

Com base na RPGA e nos recursos dos SGBDG, apresentamos um modelo de consulta que recupera os dados do SGBDG para serem processados no CMPGrafo. Esse modelo é constituído por uma expressão formal baseada na RPGA (BONIFATI et al., 2018).

Definição 13: Modelo de Consulta em Grafo (MCG). Esse modelo segue o seguinte formato:

$$\bowtie^{\phi E_{VA}}_{pos_i, pos_j \dots} \left(\bowtie^{\phi E_{sub}}_{pos_y} (r_{relsub}) \right), \text{ onde}$$

- ϕE_{VA} – é um expressão que combina as formas de seleção por dimensão, por

propriedade e por relacionamento para recuperar os Vértices Analíticos e os Vértices Dimensionais relacionados. Dessa forma, ϕE_{VA} é uma conjunção de expressões que seguem os seguintes termos:

- $\lambda(pos) = r \mid pos \in \{src_1, trg_1, \dots, src_n, trg_n\}\ and\ r \in$ RM;
- $pos_i, pos_j \mid pos_i, pos_j \in \{src_1, trg_1, \dots, src_n, trg_n\}\ \wedge\ src_i \sqsubseteq$ VD $\wedge\ trg_j \sqsubseteq$ VA;
- $pos_i = pos_j \mid pos_i, pos_j \in \{src_1, trg_1, \dots, src_n, trg_n\}$;
- $pos.p\ \theta\ val \mid pos \in \{src_1, trg_1, \dots, src_n, trg_n, edge_1 \dots, edge_n\}, \{p\} \in KM$, $val \in NM, \theta \in \{=, \neq, <, >, \leq, \geq\}$; e
- $\psi(pos)\ \theta\ val\ or\ \psi(pos.p)\ \theta\ val \mid \{p\} \in KM, val \in NM, \psi \in \{COUNT, SUM, AVG\}, pos \in \{src_1, trg_1, \dots, src_n, trg_n, edge_1 \dots, edge_n\}, \theta \in \{=, \neq, <, >, \leq, \geq\}$;

- ϕE_{sub} – é uma expressão para recuperar os Vértices dos Subgrafos Analíticos utilizando os Vértices Analíticos que foram recuperados na expressão ϕE_{VA}. Com isso, ϕE_{sub} é uma conjunção de expressões com os seguintes termos:
  - $pos_j, pos_y \mid pos_j, pos_y \in \{src_1, trg_1, \dots, src_n, trg_n\} \wedge src_j \sqsubseteq$ VA $\wedge\ trg_y \sqsubseteq subG$; e
  - $pos_j = pos_y \mid pos_j, pos_y \in \{src_1, trg_1, \dots, src_n, trg_n\}$.

Esse modelo de consulta é uma representação formal que consiste em uma consulta aninhada. Essa consulta começa fazendo uma junção entre os VD e VA que atendem as especificações da consulta e possuem relacionamentos dimensionais (reldi) entre eles. Nessa junção, os rótulos dos VD são especificados na consulta e os rótulos dos VA são reservados na modelagem. Além disso, os VA podem ter relacionamentos analíticos (relan) que os relacionam entre si formando um grafo de fatos na modelagem. Em seguida, realiza uma junção entre os VA recuperados e os Vértices do Subgrafo Analítico (SubV), considerando um relacionamento de subgrafos (relsub) entre eles. Esse relacionamento também possui um rótulo reservado na modelagem. Com essa consulta aninhada, são recuperadas sequências de relacionamentos que relacionam VD com VA e VA com SubV, formando um padrão de resultado do MCG.

Mediante essa consulta aninhada é possível recuperar quaisquer Componentes Analíticos utilizando parâmetros e expressões na consulta, não precisando se ater à estrutura do Subgrafo Analítico. Para isso acontecer, o desenvolvedor precisa reservar alguns rótulos no Modelo MPGrafo, que detalharemos a seguir.

5.3.2 Rótulos reservados no Modelo MPGrafo

Durante a definição do Modelo MPGrafo, destacamos a necessidade de alguns elementos terem rótulo reservado para a realização de consulta. Dessa forma, detalhamos a seguir esses rótulos:

- r_{VD} - é um rótulo reservado para rotular cada vértice dimensional VD_i que compõe a mesma Dimensão D, tal que $D(r_{VD}) = \bigcup_{i>0/n} VD_i(r_{VD})$, onde n é a quantidade de VD que compõe a dimensão D e r_{VD} é a nomenclatura para o rótulo de todos os VD de uma mesma dimensão;
- r_{VA} - é um rótulo reservado para todos os Vértices Analíticos VA_i de uma modelagem, tal que $\forall i, VA_i(r_{VA})$, onde r_{VA} é a nomenclatura do rótulo de todos os VA de uma modelagem; e
- r_{relsub} - é um rótulo reservado para todos os relacionamentos de subgrafo $relsub_i$ de uma modelagem, tal que $\forall i, relsub_i(r_{relsub})$, onde r_{relsub} é a nomenclatura do rótulo de todos os relsub de uma modelagem.

Com a padronização das consultas por meio de um modelo de consulta em grafo, as respostas também passam a seguir um padrão de recuperação de informação, estabelecendo a mesma sequência de relacionamentos para diferentes modelagens. Desse modo, definimos um padrão de resultado para a representação visual em grafo agregado das consultas que seguem o MCG.

5.3.3 Padrão de resultado das consultas

A Figura 35 mostra o padrão de resultado identificando os elementos {Vértice Dimensional, Vértice Analítico, Subgrafo Analítico, Relacionamento de Subgrafo, Relacionamento Analítico, Relacionamento Dimensional} que são comuns nas respostas de qualquer consulta que atendem ao MCG.

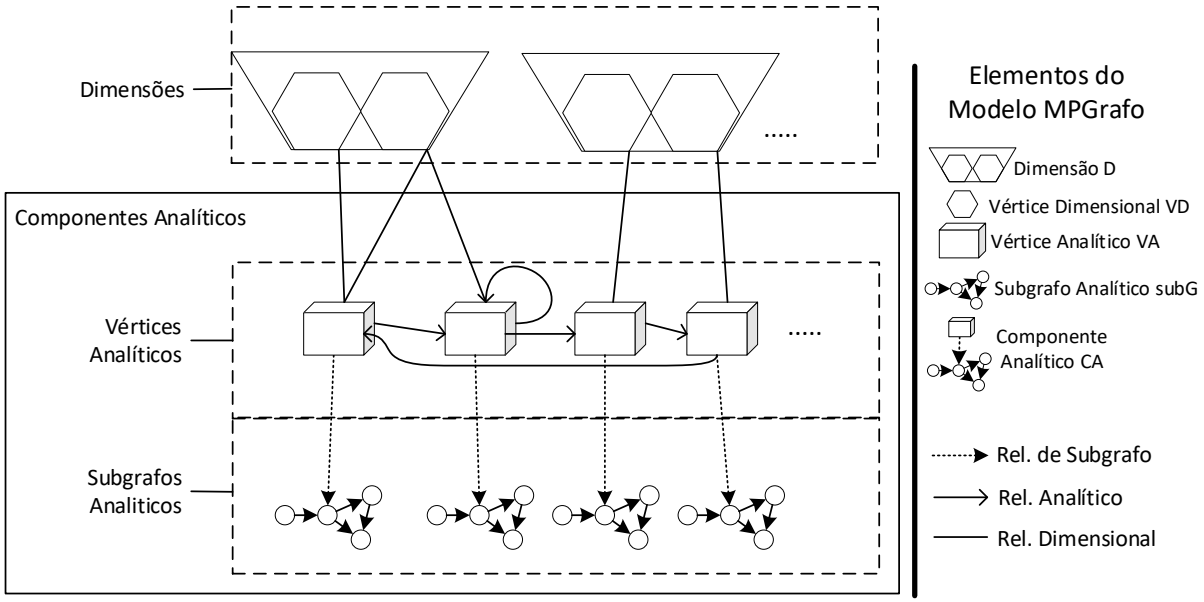


Figura 35 - Padrão de resultado do MCG

Com a definição do modelo de consulta e do padrão de resultado, apresentamos a seguir exemplos de consultas sobre duas modelagens da amostra do DBLP, que atendem ao Modelo MPGrafo. A primeira modelagem MPGrafo-1 foi definida para analisar apenas um vértice no subgrafo e a segunda modelagem MPGrafo-2 foi definida para analisar um conjunto de vértices no subgrafo de dados. Com essas modelagens, apresentamos exemplos de consulta utilizando o MCG e ilustrações para retratar a informação recuperada. Exemplos extras são expressos no Apêndice A.

5.3.4 Exemplos de consultas na modelagem MPGrafo-1

Nos exemplos dessa seção, consideramos uma modelagem que visa analisar os autores por meio de consultas sobre os vértices do tipo (:Person). Na formação dessa modelagem, utilizamos um Meta Grafo MG1 = { Rv | λ(Rv) = :Person} para especificar a estrutura dos Subgrafos Analíticos. A Figura 36 apresenta uma representação da modelagem MPGrafo-1, que atende ao MG1 e possui um grafo de fatos, o qual retrata os relacionamentos de “coauthor” entre os vértices (:Person). Essa modelagem possui três Dimensões para especificar as características dos autores (:Person), indicando o gênero (DimGender), os tipos de publicações realizadas (DimPublicationType) e as datas de publicação dos trabalhos (DimDate). Na formação da modelagem, cada Vértice Analítico foi composto por um identificador e duas propriedades: “degreeM” que informa o grau de centralidade do vértice (:Person) na modelagem, e “salary” que informa o salário do vértice (:Person) do subgrafo.

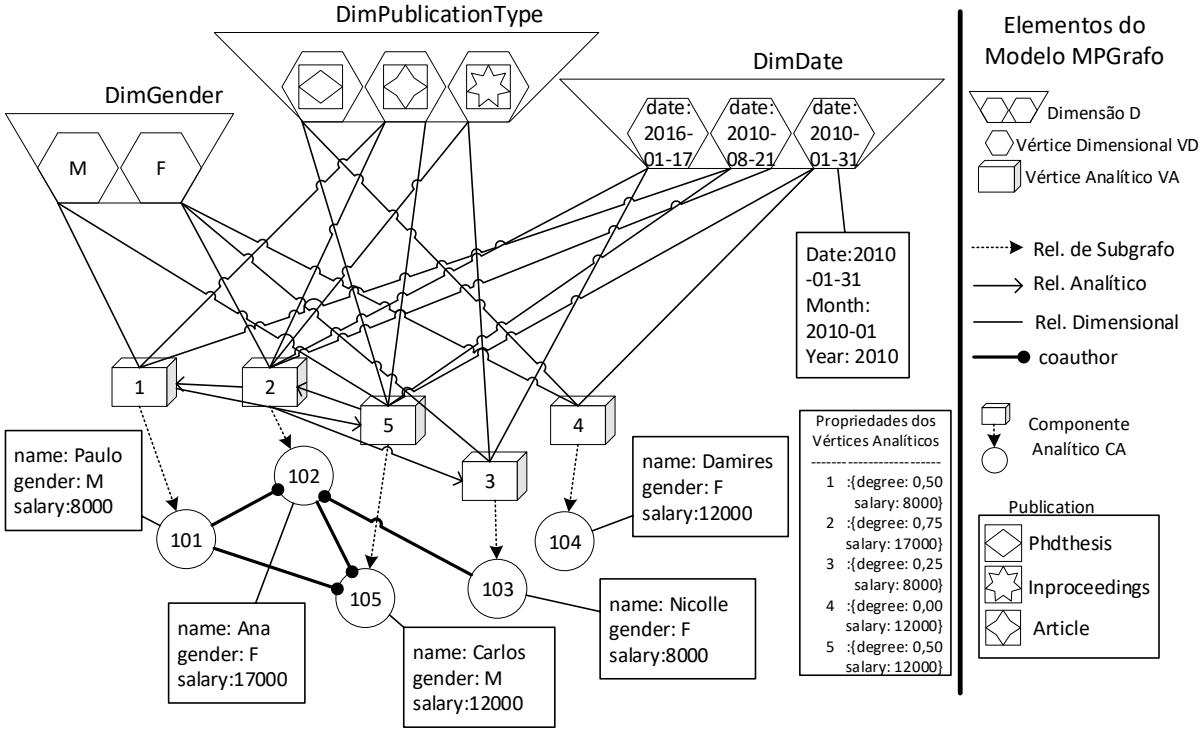


Figura 36 - Modelagem MPGrafo-1 com um vértice (:Person) no subgrafo

Com base no modelo de consulta em grafo, apresentamos a seguir exemplos de consultas que utilizam r_{VA} para o rótulo dos vértices analíticos e r_{relsub} para o rótulo dos relacionamentos de subgrafo.

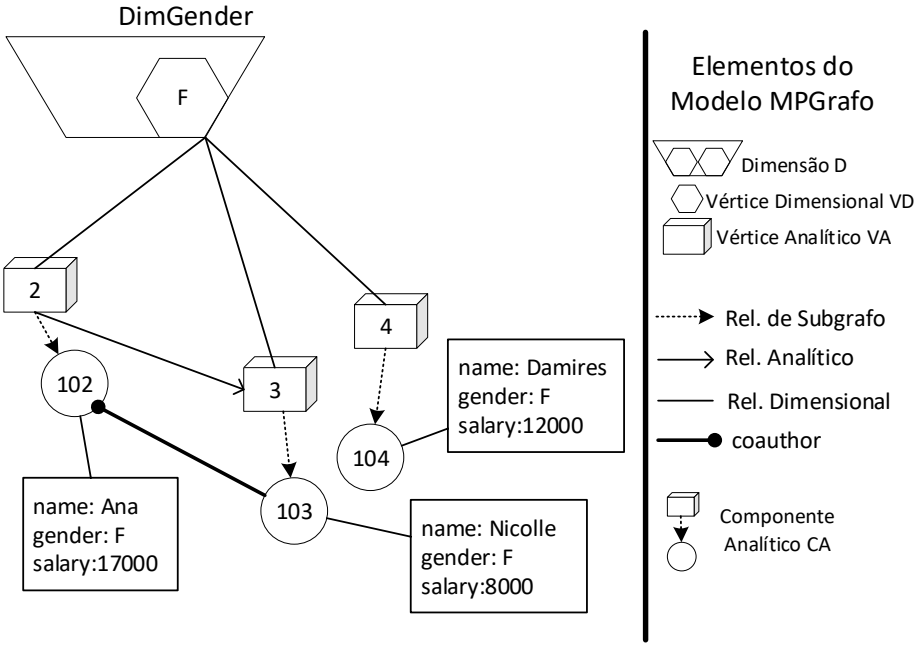


Figura 37 - Consulta os autores do gênero feminino

A Figura 37 mostra o resultado da consulta dos autores do gênero feminino. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,\ trg_1} \left(\bowtie^{\phi E_{sub}}_{src_2,\ trg_2} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimGender ∧ src_1.gender = “F” ∧ $\lambda\ (trg_1) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_2 \wedge src_2 = trg_2 \wedge \lambda\ (src_2) = r_{VA}$

Nessa consulta, ϕE_{VA} requisita os relacionamentos entre os vértices src_1 de rótulo “:DimGender” com o valor “F” na propriedade ‘gender’ e os vértices trg_1 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 são os vértices src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$ e os vértices src_2 possuem relacionamentos entre si

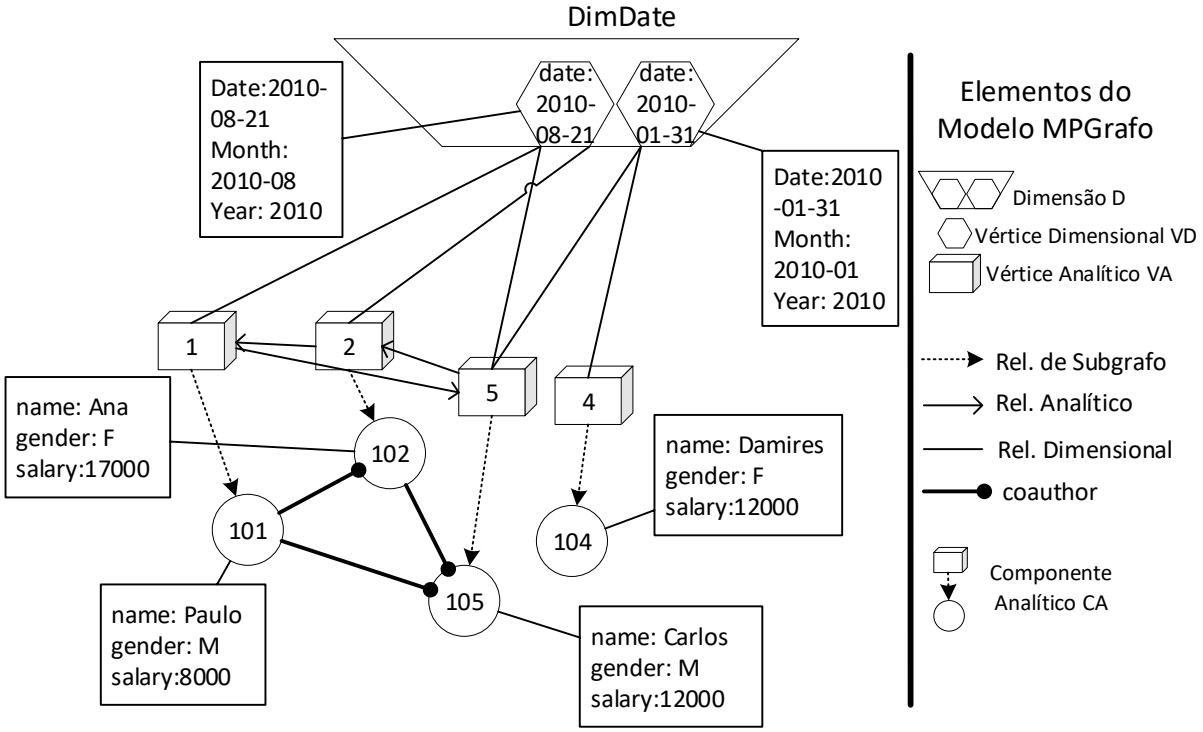


Figura 38 - Consulta os autores que publicaram em 2010

A Figura 38 mostra o resultado da consulta dos autores que publicaram em 2010. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,\ trg_1} \left(\bowtie^{\phi E_{sub}}_{src_2,\ trg_2} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate ∧ src_1.year = 2010 ∧ $\lambda\ (trg_1) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_2 \wedge src_2 = trg_2 \wedge \lambda\ (src_2) = r_{VA}$

Nessa consulta, ϕE_{VA} requisita os relacionamentos entre os vértices src_1 de rótulo “:DimDate” com o valor “2010” na propriedade ‘year’ e os vértices trg_1 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$ e os vértices src_2 possuem

relacionamentos entre si.

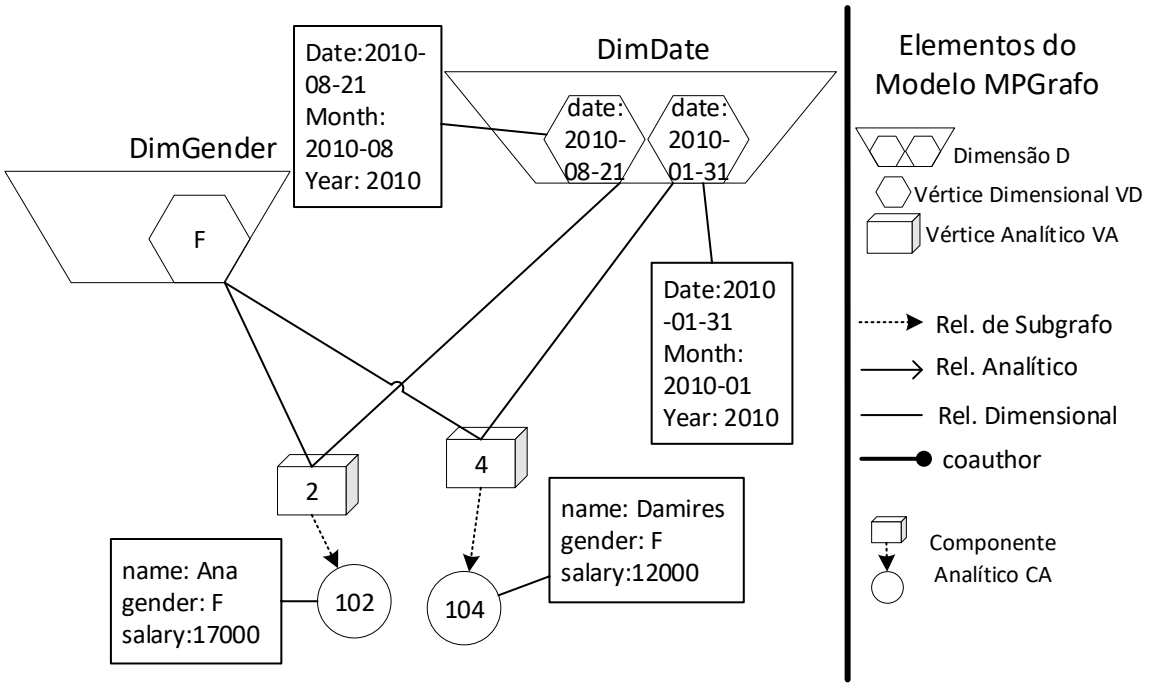


Figura 39 - Consulta os autores que publicaram em 2010 e são do gênero feminino.

A Figura 39 mostra o resultado da consulta dos autores do gênero feminino que publicaram em 2010. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{trg_3} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\,(src_1)$ = :DimDate $\wedge$ src_1.year = 2010 $\wedge$ $\lambda\,(trg_2) = r_{VA} \wedge \lambda\,(src_2)$ = :DimGender $\wedge$ src_2.gender = "F" $\wedge$ $trg_2 = trg_1$
- ϕE_{sub}: $trg_2 = src_3$

Nessa consulta, ϕE_{VA} requisita dois tipos de relacionamentos: o primeiro, entre os vértices src_1 de rótulo ":DimDate" com o valor "2010" na propriedade 'year' e os vértices trg_1 de rótulo r_{VA}; e o segundo, entre os vértices src_2 de rótulo ":DimGender" com o valor "F" na propriedade 'gender'. Além disso, trg_1 e trg_2 correspondem aos mesmos vértices, sendo representados por trg_2 na consulta. ϕE_{sub} especifica que os vértices encontrados trg_2 são representados por src_3 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_3 \in subV$.

5.3.5 Exemplos de consultas na modelagem MPGrafo-2

Nos exemplos de consulta dessa seção utilizaremos uma modelagem que analisa um conjunto de vértices (subgrafo) com informações de publicação. Na produção da modelagem, utilizamos o Meta Grafo MG2 = { $(Rv_1)^+$ -[Ra_{12}]-› Rv_2 ‹-[Ra_{23}]- Rv_3 | $\lambda(Rv_1)$ = :Person $\wedge$ $\lambda(Rv_2)$ = :Publication $\wedge$ $\lambda(Rv_3)$ = :Venue $\wedge$ $\lambda(Ra_{12})$ = :author $\wedge$ $\lambda(Ra_{23})$ = :publish

} para especificar o padrão do grafo nos subgrafos analíticos. Nesse MG2 os subgrafos podem ter um ou mais vértices Rv_1, com rótulo (:Person), se relacionando com um vértice Rv_2, com rótulo (:Publication), e, em seguida, esse vértice Rv_2 se relacionando com um vértice Rv_3, com rótulo (:Venue).

Com essa especificação, definimos a modelagem MPGrafo-2 que atende ao MG2 e possui três Dimensões para caracterizar os Componentes Analíticos. As Dimensões detalham os autores da publicação (DimAuthor), os locais de publicação (DimPublicationVenue) e as datas de publicação (DimDate).

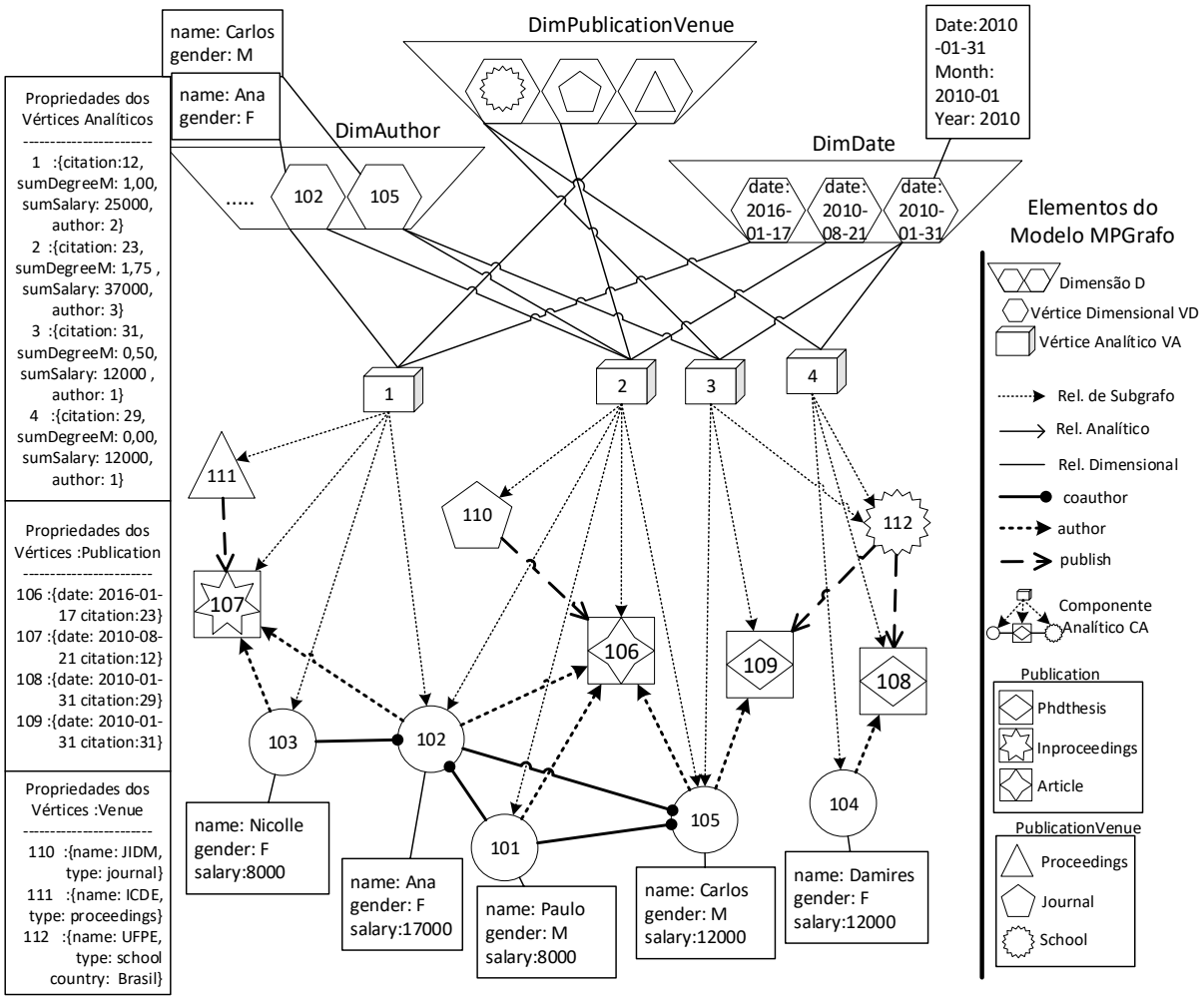


Figura 40 - Modelagem MPGrafo-2 contendo um conjunto de vértices no subgrafo

Além das dimensões, valores de propriedades foram pré-processados para o cálculo das medidas de análise na consulta. Nessa modelagem, foram definidas no Vértices Analíticos as seguintes propriedades: ‘sumSalary’, resultado da soma de todos os salários das pessoas que compõem o subgrafo; ‘sumDegreeM’, resultado da soma do grau de centralidade das pessoas que compõe o subgrafo, ‘author’ número de vértices (:Person) contido no subgrafo e ‘citation’, obtido da propriedade do vértice (:Publication) que compõe

o subgrafo. Com essa descrição, apresentamos na Figura 40 a modelagem MPGrafo-2, contendo as três dimensões e os valores pré-processados nos VA.

Apresentamos a seguir exemplos de consulta na modelagem MPGrafo-2 utilizando r_{VA} para o rótulo dos vértices analíticos e r_{relsub} para o rótulo dos relacionamentos de subgrafo.

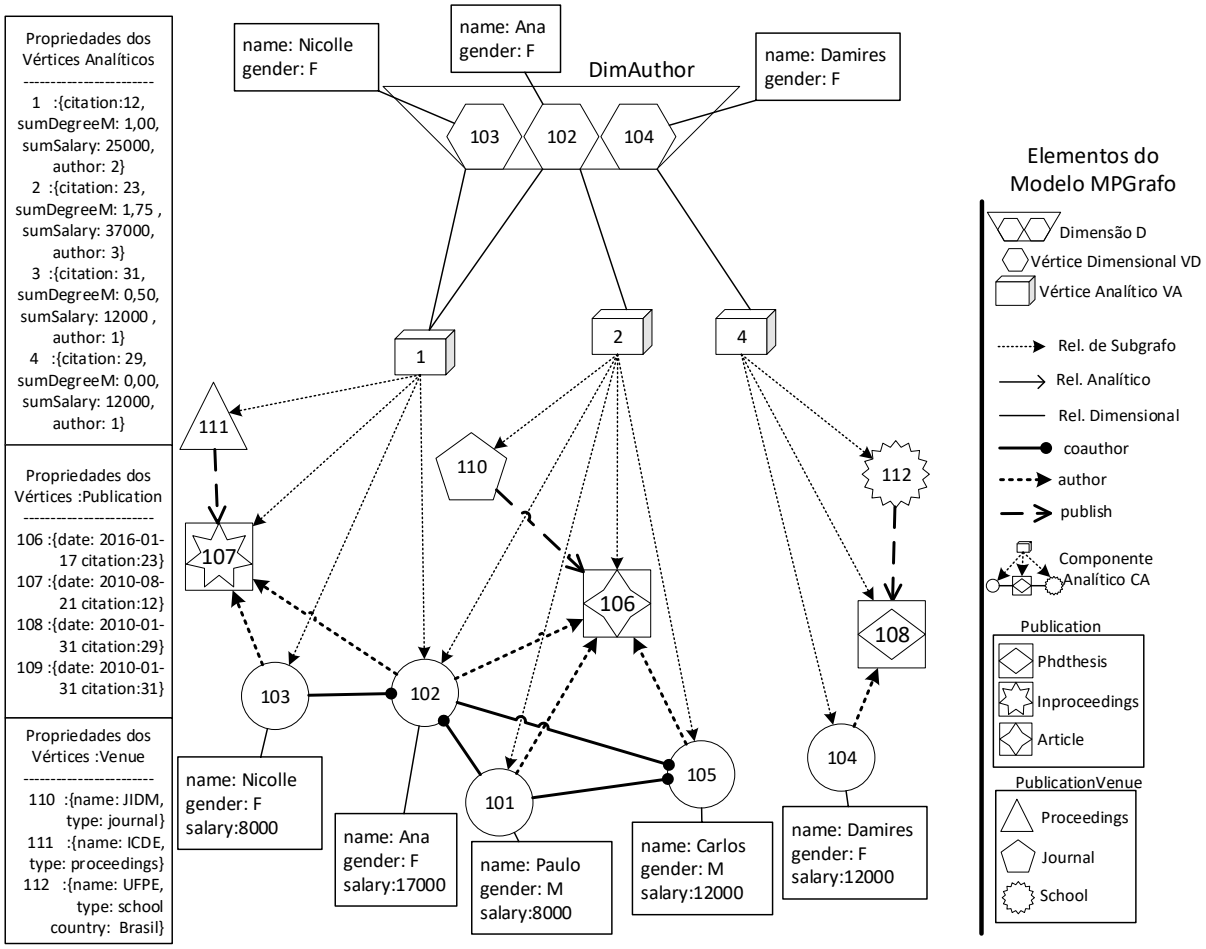


Figura 41 - Consulta as publicações que possuem autores do gênero feminino

A Figura 41 mostra o resultado da consulta que requisita as informações de publicação que possuem autores do gênero feminino. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimAuthor $\wedge\ src_1$.gender = F $\wedge\ \lambda\ (trg_1) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_2$

Nessa consulta, ϕE_{VA} requisita os relacionamentos entre os vértices src_1 de rótulo ":DimAuthor" com o valor "F" na propriedade 'gender' e os vértices trg_1 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$. Observe que essa consulta

é semelhante à consulta da Figura 37, mostrando um resultado específico da modelagem MPGrafo-2.

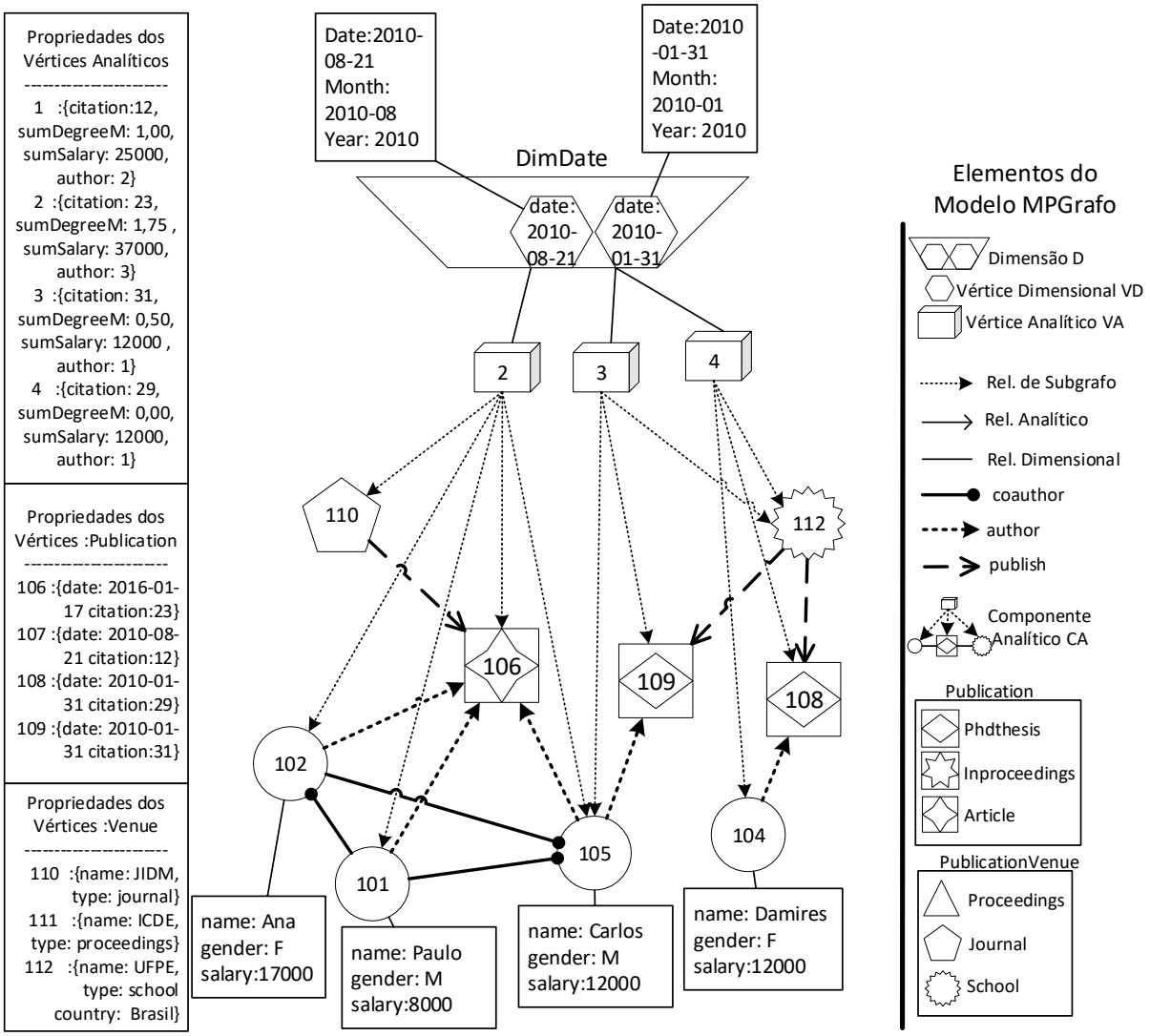


Figura 42 - Consulta as publicações de 2010.

A Figura 42 mostra o resultado da consulta que requisita as informações de publicações de 2010. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge$ src_1.year = 2010 $\wedge$ $\lambda\ (trg_1) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_2$

Nesse exemplo, ϕE_{VA} requisita o relacionamento entre os vértices src_1 de rótulo ":DimDate" com o valor "2010" na propriedade 'year' e os vértices trg_1 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$. Observe que essa consulta é equivalente à consulta da Figura 38.

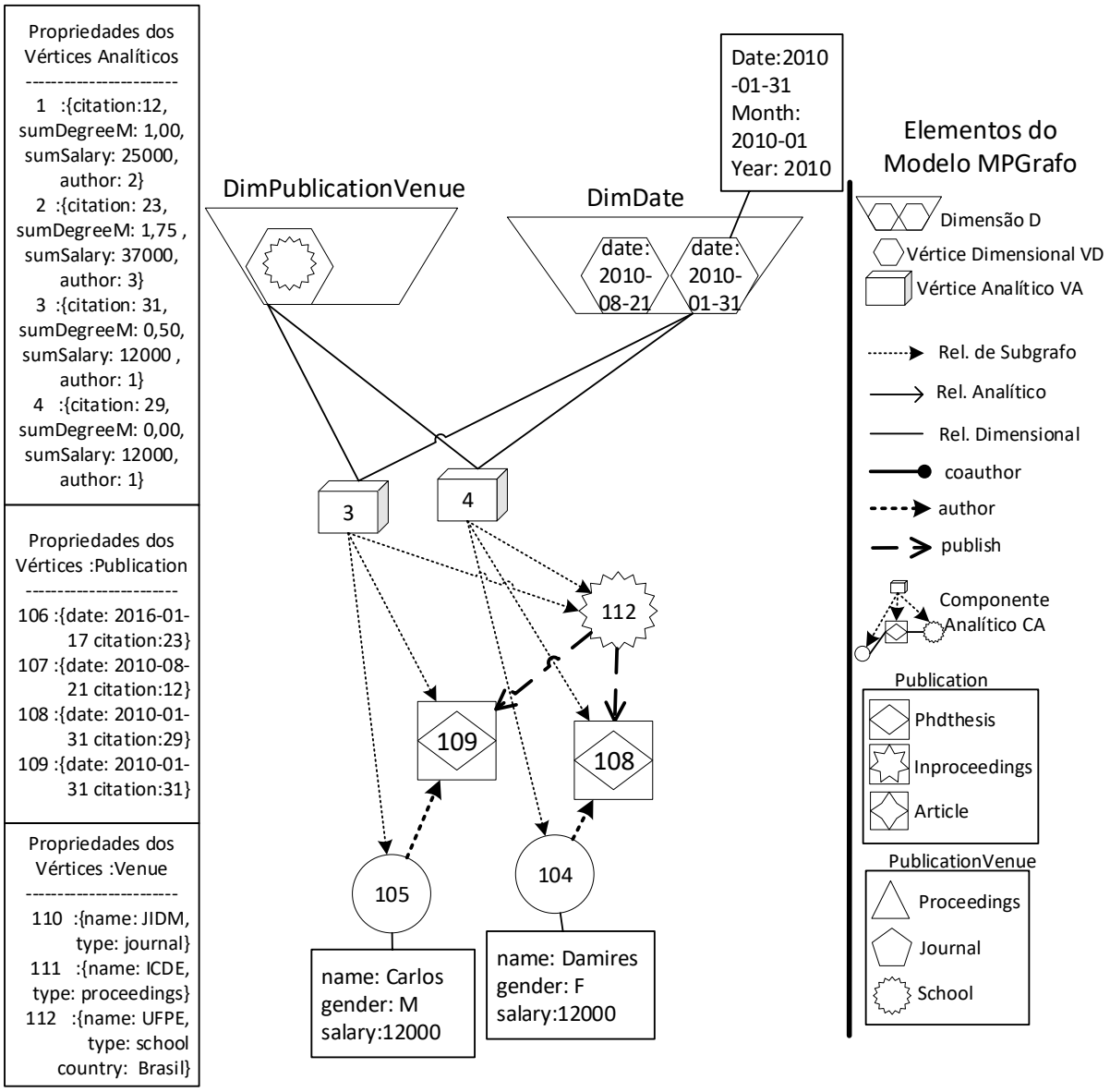


Figura 43 - Consulta as publicações das instituições que foram realizadas em 2010

A Figura 43 mostra o resultado da consulta que requisita as informações de publicações das instituições que foram realizadas em 2010. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{trg_3} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge$ src_1.year = 2010 $\wedge$ $\lambda\,(trg_2) = r_{VA} \wedge \lambda\ (src_2)$ = :DimPublicationVenue $\wedge$ src_2.type = “School” $\wedge$ $trg_2 = trg_1$
- ϕE_{sub}: $trg_2 = src_3$

Nessa consulta, ϕE_{VA} requisita dois tipos de relacionamentos: o primeiro, entre os vértices src_1 de rótulo “:DimDate” com o valor “2010” na propriedade ‘year’ e os vértices trg_1 de

rótulo r_{VA}; e o segundo, entre os vértices src_2 de rótulo ":DimPublicationVenue" com o valor "School" na propriedade 'type' e os vértices trg_2 de rótulo r_{VA}. Além disso, trg_1 e trg_2 correspondem aos mesmos vértices, sendo representados por trg_2 na consulta. ϕE_{sub} especifica que os vértices encontrados trg_2 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_3 \in subV$.

5.3.6 Parâmetro de Configuração Dimensional

Nos exemplos das Figuras 38 e 42, as consultas especificam que os Vértices Dimensionais VD da dimensão ":DimDate" tenham o valor "2010" na propriedade "year". Com essa especificação é recuperado mais de um VD da mesma dimensão que atendam a essas características. O retorno de vários VD com as mesmas características prejudica o processamento de agregação, pois os Vértices Analíticos VA são agregados em função dos Vértices Dimensionais. Dessa maneira, teriam diferentes agregações de VA com as mesmas características de VD, o que torna errada a agregação. Além disso, existe uma limitação na seleção por dimensão, que impede de requisitar os Vértices Dimensionais na consulta sem fixar um valor na propriedade. Essa limitação, impossibilita a agregação por meio dos valores existentes na propriedade dos VD.

Para corrigir a agregação e acabar com essa limitação, definimos um parâmetro, denominado Configuração Dimensional, para informar os rótulos e as propriedades dos VD que serão unificados durante a consulta. Essa unificação consiste em definir para cada tupla de característica { r_{VD}, k_{VD}, n_{VD}} um Vértice Dimensional Representante VDr, onde r_{VD} é o rótulo da dimensão, $k_{VD} \in$ KD é a propriedade chave, e $n_{VD} \in$ NM é o valor.

Definição 14: Vértice Dimensional Representante (VDr) é um vértice criado durante a consulta para representar os vértices dimensionais que possuem as mesmas características destacadas na consulta. Com isso, o VDr passa a representar essas características integrando os relacionamentos dimensionais de todos os VD recuperados.

Na unificação dos VD, a configuração dimensional informa a combinação de rótulo e propriedade que os VD precisam seguir e para cada valor da propriedade informada é criado um VDr. Com isso, os VDr representam os VD que possuem a mesma tupla de características, fazendo com que todos os relacionamentos dimensionais dos VD sejam realizados pelos VDr. A Figura 44 apresenta as mudanças da consulta na Figura 38 ao utilizar o parâmetro de configuração dimensional.

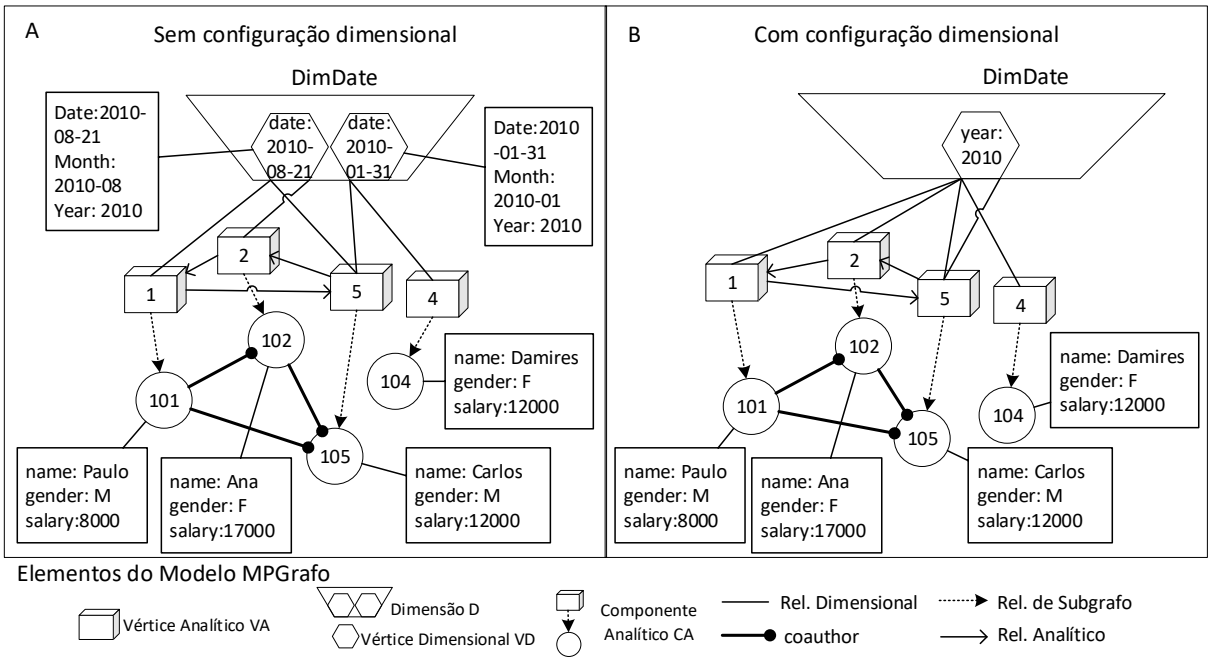


Figura 44 - Diferença da consulta considerando a configuração dimensional

Como pode ser observado, o resultado da Figura 44A apresenta dois vértices dimensionais e da Figura 44B um vértice dimensional. Essas duas imagens especificam as mesmas características na consulta, mas a imagem B realizaria corretamente a agregação, pois todos os VA, que possuem as mesmas características estão relacionados ao mesmo VD. Portanto, o parâmetro de Configuração Dimensional auxilia as consultas em SGBDG permitindo funcionalidades de consultas OLAP.

5.3.7 Operadores OLAP

Na AAMPGrafo, a aplicação de operadores OLAP consiste em realizar consultas parametrizadas que mostrem resultados equivalentes às operações OLAP tradicionais.

5.3.7.1 Operação Roll-up/ Drill-down

Para realizar operações do tipo Roll-up/ Drill-down, a modelagem precisa ter propriedades que expressam níveis hierárquicos nos Vértices Dimensionais. Por exemplo, a dimensão DimDate que define uma hierarquia por meio das propriedades year → month → date. Além disso, as consultas precisam especificar quais propriedades dos VD serão consideradas na recuperação dos dados para realizar a agregação. Dessa maneira, mediante a especificação da propriedade é definido o nível hierárquico na agregação dos dados. Para melhorar a compreensão, isolamos a dimensão hierárquica DimDate da modelagem MPGrafo-1, como mostra a Figura 45.

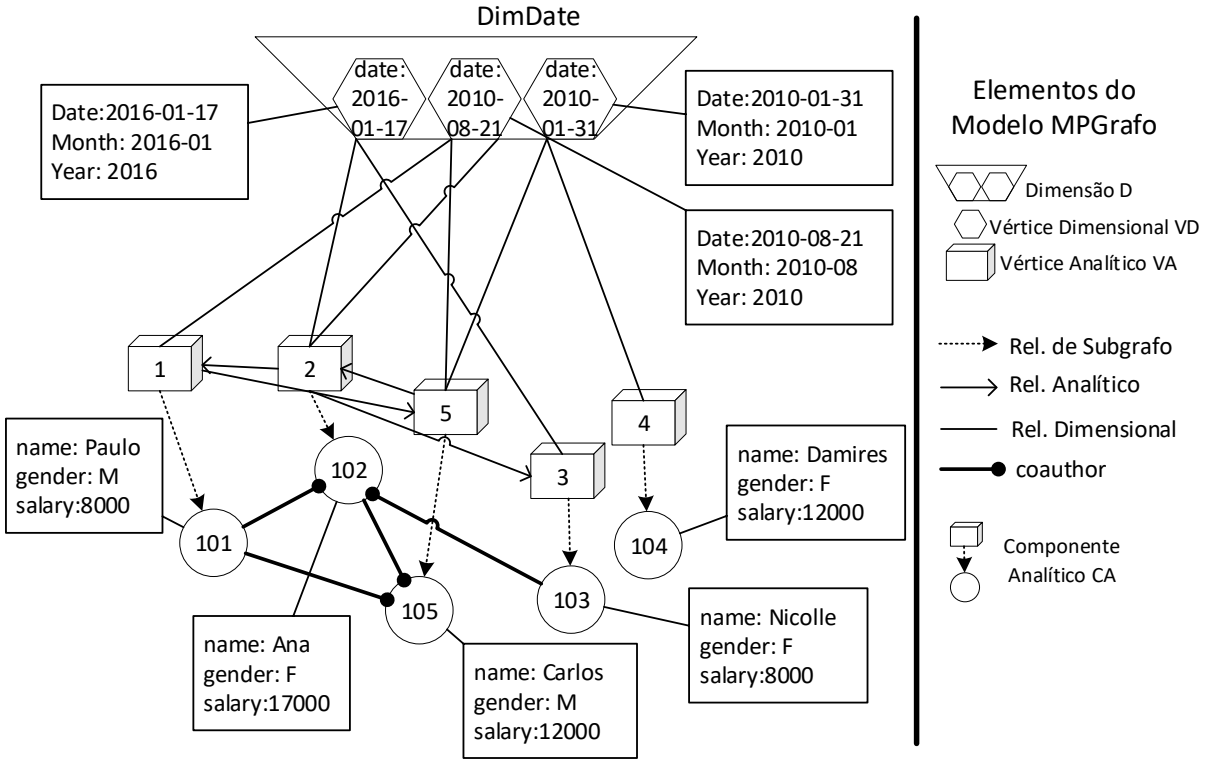


Figura 45 - Modelagem MPGrafo-1 com a dimensão DimDate isolada

Na modelagem MPGrafo-1, a formação da dimensão :DimDate consistiu em criar um único VD para cada valor de propriedade “date”, que é considerada a informação de menor granularidade dessa dimensão. Além disso, as propriedades year e month foram definidas nos VD, com o intuito de qualificar as informações de date.

A Figura 45 mostra uma modelagem retratando o menor nível hierárquico da dimensão “:DimDate”. Com isso, apresentamos a seguir um exemplo de consulta que realiza a operação Roll-up sobre a modelagem da Figura 45.

A Figura 46 mostra o resultado da consulta que agrega os autores considerando os anos de publicação. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge\ \lambda\ (trg_1) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_2$
- Configuração Dimensional: { DimDate:[year] }

Nessa consulta, ϕE_{VA} requisita os relacionamentos entre os vértices src_1 de rótulo “:DimDate” e os vértices trg_1 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$. Após a execução dessa consulta, os VD recuperados são unificados, considerando o rótulo “DimDate” e os valores existentes na propriedade “year”, como ilustra a Figura 46.

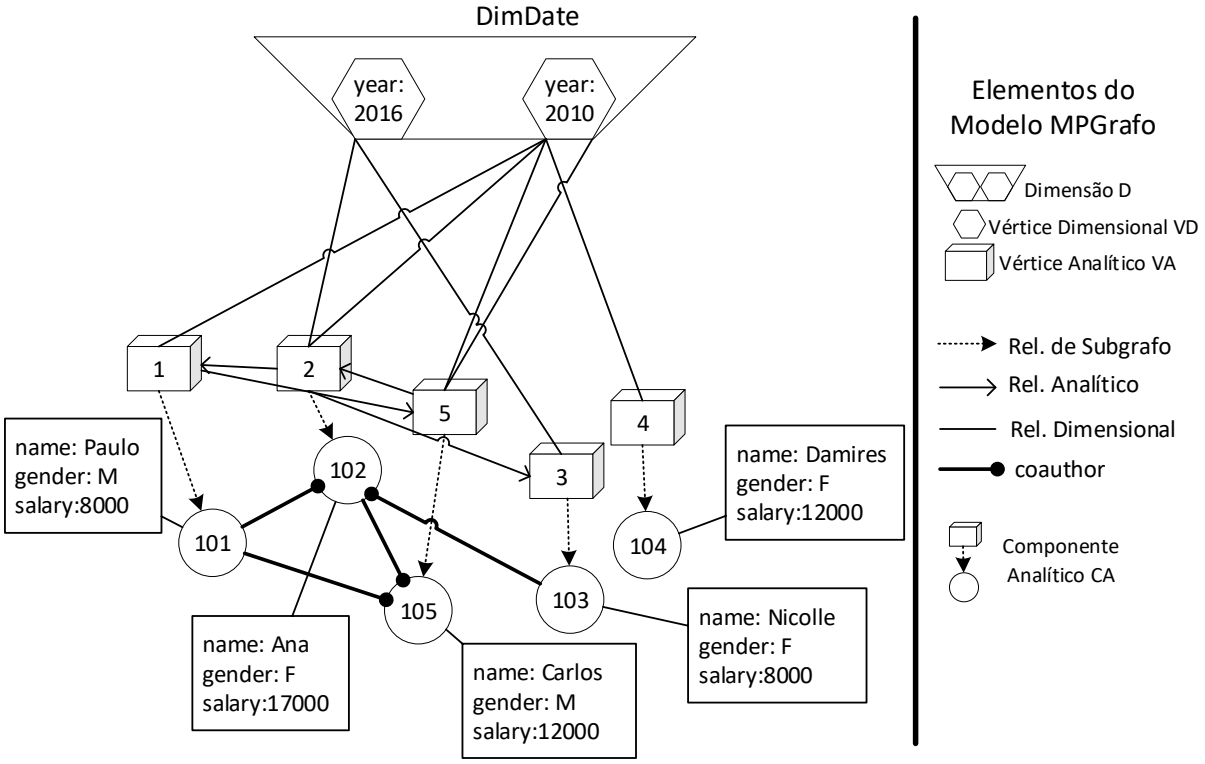


Figura 46 - Consulta e agrega os autores em função dos anos de publicação

Apresentamos uma consulta que realiza a operação Drill-down sobre o resultado da Figura 46 no Apêndice B. Observando as consultas, percebemos que as operações Roll-up/Drill-down dependem do parâmetro de Configuração Dimensional, podendo realizar Roll-up ou Drill-down em função do nível hierárquico da propriedade especificada.

5.3.7.2 Operação Slice/Dice

Para realizar operações do tipo Slice/Dice, a modelagem MPGrafo precisa ter dimensões para especificar quais características considerar na visualização dos dados. O Slice seleciona os dados considerando as características de uma única dimensão. Essa operação já foi abordada nos exemplos das Figuras 38 e 42, nas quais restringe a visualização considerando o ano de 2010. O Dice seleciona os dados considerando as características de duas ou mais dimensões. Essa operação também foi abordada nos exemplos das Figuras 39 e 43, a qual restringe a visualização especificando valores em duas dimensões diferentes.

Concluindo a etapa de seleção, observamos que o modelo de consulta em grafo, os rótulos reservados e o parâmetro de Configuração Dimensional permitem recuperar os dados da modelagem MPGrafo, independente das informações dos Subgrafos Analíticos. Além disso, a combinação desses recursos permitem realizar operações OLAP e consultas

multidimensionais em grafo de forma genérica, recuperando os dados para serem processados nas próximas etapas do modelo de CMPGrafo.

5.4 ETAPA DE ANÁLISE TOPOLÓGICA

A etapa de Análise Topológica consiste em receber os dados recuperados da etapa de seleção para analisar a estrutura topológica do grafo de fatos constituída por relacionamentos analíticos e Vértices Analíticos. Essa etapa de processamento considera o parâmetro de configuração topológica na especificação da análise em grafo. Consequentemente, a etapa de análise topológica é encarregada de processar apenas a estrutura topológica que compõe os Vértices Analíticos, desconsiderando a topologia do subgrafo. Dessa forma, a estrutura topológica é alterada de acordo com as dimensões e as especificações da consulta, podendo processar no grafo de fatos diferentes algoritmos de análise.

O sequenciamento das atividades no CMPGrafo faz com que as etapas sejam independentes. Dessa forma, a etapa de análise topológica pode integrar quaisquer dos algoritmos de análise em grafo que possa analisar os Vértices Analíticos. Com isso, definimos nessa primeira versão da AAMPGrafo medidas tradicionais de análise em grafo, tais como: *closeness*, *betweenness* e variações do *degree centrality*.

- *Closeness* – é um algoritmo que calcula uma pontuação para os vértices do grafo, avaliando a aproximação de um vértice com os outros vértices do grafo. No processamento desse algoritmo os vértices centrais adquirem as maiores pontuações, considerado que quanto menos distante um vértice é dos outros vértices, menor será a soma da distância geodésica (caminho mais curto). O algoritmo é definido pelo inverso da soma da distância geodésica de $v \in$ Va para todos os outros Vértices Analíticos alcançáveis a partir de v:

$$C_C(v) = \frac{1}{\sum_{t \in VA \setminus v} d_G(v,t)}$$

onde $t \in Va$ e $\mathrm{d_G}(v,t)$ é a distância geodésica dos vértices t alcançáveis por v.

- *Betweenness* – é uma medida de centralidade definida pela ocorrência de um Vértice Analítico nos menores caminhos formados dentro do grafo pelos outros Vértices Analíticos. Esta medida valoriza os vértices que possibilitam os menores caminhos entre vértices. Para um grafo com n vértices, o cálculo do *betweenness* precisa:

4. Determinar para cada par de vértices (α, μ) a quantidade de menores caminhos entre eles;
5. Determinar para cada par de vértices (α, μ) a fração de menores caminhos que passam sobre o vértice v e todos os menores caminhos entre eles;
6. Somar todas as frações dos pares de vértices (α, μ).

O algoritmo é representado pela seguinte fórmula:

$$C_B(v) = \sum_{\alpha \neq v \neq \mu \in VA} \frac{\sigma_{\alpha\mu}(v)}{\sigma_{\alpha\mu}}$$

onde $\sigma_{\alpha\mu}$ é a quantidade de menores caminhos de α para μ e $\sigma_{\alpha\mu}(v)$ é o número de menores caminhos de α para μ que passam pelo vértice v.

- *Degree centrality* – é uma medida que considera o número de ligações incidentes sobre um vértice, ou seja, o número de ligações que um vértice contém. Com base nessa especificação, definimos variações dessa medida que são apresentadas a seguir:
    - *indegree* – conta o número de ligações direcionadas ao vértice $\mathrm{indeg}\,(v)$ em proporção ao número de vértices menos 1;

$$C_{indegree}(v) = \frac{\mathrm{indeg}\,(v)}{n-1}$$

    - *outdegree* – conta o número de relações direcionadas de um vértice aos outros vértices $\mathrm{outdeg}\,(v)$ em proporção ao número de vértices menos 1;

$$C_{outdegree}(v) = \frac{\mathrm{outdeg}\,(v)}{n-1}$$

    - *degree* – conta o número de relações para um vértice $\mathrm{deg}\,(v)$ em proporção ao número de vértices menos 1;

$$C_{degree}(v) = \frac{\mathrm{deg}\,(v)}{n-1}$$

  Além dos diferentes cálculos nas variações, cada variação pode especificar quais rótulos de relacionamentos são permitidos no cálculo dessas medidas.

Nessa etapa, após o processamento dos algoritmos, os Vértices Analíticos recebem propriedades com valores que representam o resultado da análise de cada algoritmo requisitado na consulta. Com isso, a etapa de análise topológica define valores da análise topológica para os Vértices Analíticos, de modo que não altera a estrutura do padrão de

resposta da etapa de seleção.

5.5 ETAPA DE AGREGAÇÃO E VISUALIZAÇÃO

A etapa de agregação e visualização consiste em receber os dados das etapas anteriores e realizar a agregação, definindo uma visualização gráfica que corresponda à resposta da consulta. Para isso, definimos dois algoritmos para agregar os Componentes Analíticos e produzir uma representação gráfica que ilustra a resposta da consulta.

Nesses algoritmos, utilizamos um modelo em grafo de propriedade para representar a resposta das consultas e assim mostrar tanto os valores agregados quanto a estrutura topológica agregada. Os elementos desse modelo são introduzidos a seguir:

Definição 15: Vértice Analítico Agregado (VAagr) é um vértice criado durante a consulta para representar visualmente a agregação dos vértices analíticos. Esse vértice possui propriedades com valores agregados que representam os resultados das medidas de análise e a quantidade de VA que compõe a sua agregação.

Definição 16: Subgrafo Agregado (SubGagr) é uma representação em grafo de propriedade criada durante a consulta para representar visualmente a agregação de subgrafos analíticos. Esse grafo é composto por elementos agregados que são representados por vértices e arestas com propriedades para especificar a agregação. Essas propriedades são utilizadas na agregação para expressar a quantidade de elementos que compõem o elemento agregado e para registrar o resultado de funções de agregações. Dessa forma, definimos SubGagr = (SubVagr,SubAagr), onde SubGagr agrega a estrutura topológica de um conjunto de Subgrafos Analíticos SubG e acrescenta nos elementos SubGagr = (SubVagr,SubAagr) propriedades que registrem a agregação dos elementos SubG = (SubV,SubA), respectivamente.

Definição 17: Componente Analítico Agregado (CAagr) é uma representação composta por um VAagr e um SubGagr no qual o VAagr informa os valores agregados e os relacionamentos agregados entre os CAagr e o SubGagr apresenta a agregação da estrutura topológica. Com isso, o CAagr=(VAagr, SubGagr) é a representação de um fato agregado em grafo que mostra a agregação tanto dos valores quanto da estrutura topológica dos dados analisados. O CAagr é formado por meio dos relacionamentos que vinculam um VAagr a um SubGagr correspondente.

Definição 18: Modelo de Resposta em Grafo Multidimensional (MRGM). esse modelo foi definido para ajudar no entendimento das informações recuperadas na consulta multidimensional em grafo. Ele é composto por quatro partes de representação que formam a resposta da consulta. Esse modelo segue a seguinte estrutura MRGM = (VD, VAagr, SubGagr, Gd, Avd, Ava, Asub, Agd), onde

- VD ⊂ D é o conjunto de Vértices Dimensionais que foram recuperados na etapa de seleção;
- VAagr é o conjunto de Vértices Analíticos Agregados, que foram produzidos durante a consulta por meio da agregação dos VA;
- SubGagr é o conjunto de Subgrafos Agregados, que foram formados por meio das agregações dos subgrafos analíticos, os quais se relacionam com os VA que foram agregados.
- Gd é a representação do grafo de dados que contém os vértices de dados
- Avd é um conjunto de arestas com peso que relacionam os VD com os VAagr, contabilizando os relacionamentos entre eles;
- Ava é um conjunto de arestas com peso que relacionam os VAagr, contabilizando os relacionamentos entre os VAagr;
- Asub é um conjunto de arestas que relacionam os VAagr com os SubGagr correspondentes.
- Agd é um conjunto de arestas com peso que relacionam os vértices do subgrafo agregado SubVagr com os vértices V do grafo de dados.

A Figura 47 mostra um exemplo genérico do MRGM, destacando cada parte de representação.

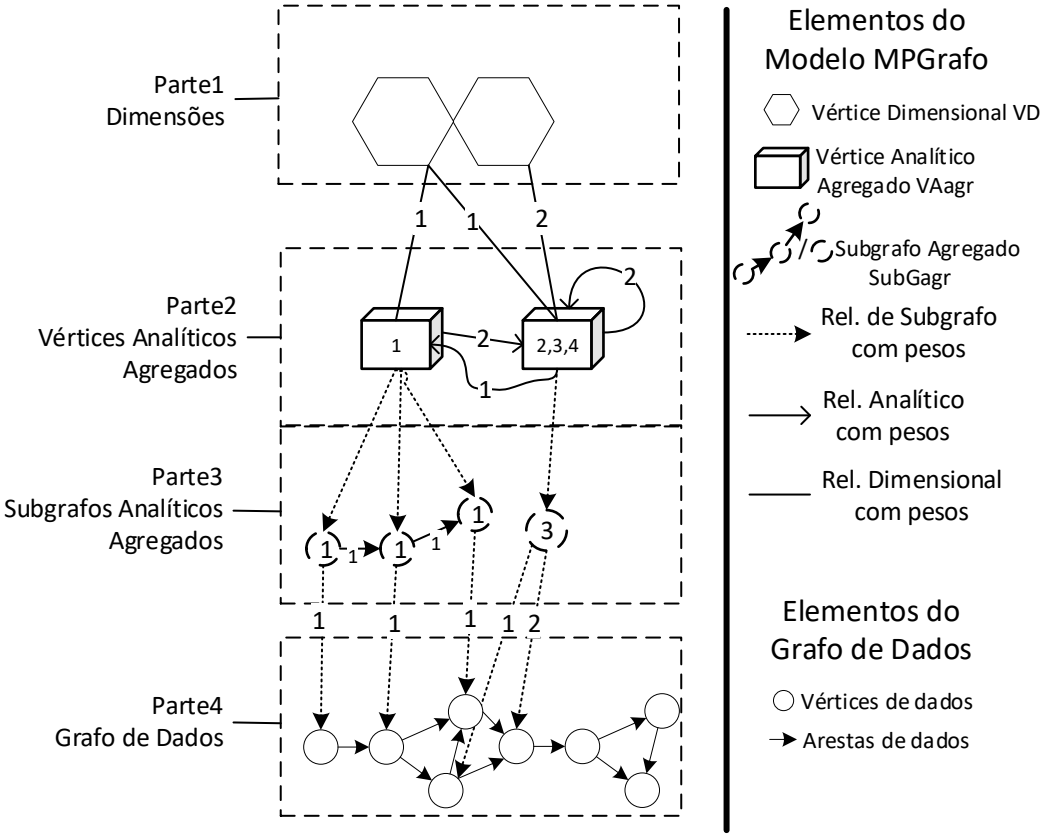


Figura 47 - Modelo de Resposta em Grafo Multidimensional

A parte 1 representa as dimensões que mostram os vértices dimensionais (VD) que foram selecionados na consulta para especificar as características dos Componentes Analíticos. Além disso, essa parte utiliza relacionamentos com pesos para indicar a quantidade de vezes que as características de um VD estão presentes em um VAagr.

A parte 2 é um conjunto de VAarg que contém valores nas propriedades para apresentar o resultado das medidas de análise. Além disso, essa parte, dependendo da Modelagem MPGrafo, pode conter relacionamentos com pesos para indicar a quantidade de vezes que um VAagr se relaciona com outro VAagr, representando, assim, os relacionamentos entre os Componentes Analíticos. Na ilustração dessa parte, cada VAagr possui os identificadores dos VA que o formaram.

A parte 3 é um conjunto de SubGagr no qual cada SubGagr é vinculado a um VAagr correspondente, compondo assim os componentes analíticos agregados. Os SubGagr são formados por meio da agregação dos subgrafos analíticos SubG. Esses SubG possuem relações com os mesmos VA que formaram o VAagr. Na ilustração dessa etapa, cada elemento que compõe os SubGagr apresenta um peso para refletir a agregação do elemento.

A parte 4 representa os dados originais do grafo de dados Gd relacionados com os

vértices SubVagr do SubGagr. Cada SubVagr é composto por um ou mais vértices do grafo de dados. Dessa forma, utilizamos uma aresta com pesos para expressar a quantidade de vezes que um vértice V do grafo de dados participa na formação do vértice agregado SubVagr do SubGagr.

Com a especificação do grafo de resposta, apresentamos a seguir os algoritmos encarregados de processar a resposta da consulta e produzir uma representação gráfica que atenda ao MRGM. Denominamos esses algoritmos de Clusterização de Componentes Analíticos (ClusCA) e Agregação dos Componentes Analíticos (AgrCA).

5.5.1 Clusterização de Componentes Analíticos (ClusCA)

A clusterização de Componentes Analíticos consiste em identificar diferentes combinações de Vértices Dimensionais que se relacionam com o mesmo Vértice Analítico. Em outras palavras, agrupar os vértices dimensionais que se relacionam com o mesmo vértice analítico. Cada grupo representa uma combinação de assunto / característica dos VD, de modo que não existem dois grupos com a mesma combinação. A seguir, cada grupo de VD é responsável por agrupar os Vértices Analíticos que atendem à combinação de assunto / característica. Com isso termina a etapa de clusterização de componentes analíticos, definindo os grupos de vértices analíticos que representam as mesmas especificidades indicadas nos vértices dimensionais. Apresentamos a seguir o pseudocódigo para descrever esse processo de clusterização.

No pseudocódigo da Figura 48, a função "v_{VA}.*findRelatedVD (VD);*" retorna um conjunto de vértices dimensionais que se relacionam com v_{VA}. Nesse algoritmo é produzido um mapeamento *map (k,v)*, onde:

- "k" é a chave da tabela que representa um conjunto (grupo) de vértices dimensionais (setVD) que se relacionam com os VA contidos em "v";
- "v" é o valor da tabela que representa um conjunto de vértices analíticos (setVA) que mantêm relacionamentos com todos os VD contidos na chave "k" correspondente.

```
ClusCA(VD,CA)
Input: VD, CA
Output: map
1     map ← Map(Set,Set);
2     VA ← CA.getVA();
3     for each v_VA in VA
4           Set k ← v_VA.findRelatedVD (VD);
5           if map.containsKey(k) then
6                 Set b ← map.get(k);
7                 b.add(v_VA);
8           else
9                 map.put(k, new Set(v_VA)));
10          end if;
11    end for;
12    return map;
13    end ClusCA;
```

Figura 48 - Algoritmo de Clusterização dos Componentes Analíticos

No final desse algoritmo, é formada uma tabela de mapeamento "*map"* que serve de entrada para a etapa de agregação dos componentes analíticos.

5.5.2 Agregação dos Componentes Analíticos (AgrCA)

A agregação dos Componentes Analíticos (AgrCA) consiste em agregar os conjuntos de Vértices Analíticos (setVA) contidos na tabela *map,* processar as medidas de análise especificadas no parâmetro configuração de medida e agregar os Subgrafos Analíticos que se relacionam com os VA do setVA.

A Figura 49 apresenta um algoritmo que recebe a Configuração de Medida e a tabela *map* para formar o grafo de resposta que atende ao MRGM. Na agregação dos dados da tabela *map* (k,v), consideramos que para cada elemento entry(k,v) da tabela *map* é definido um vértice analítico agregado VAarg, que representa a agregação do conjunto de vértices analíticos "v" e forma relacionamentos com cada vértice dimensional contido em "k".

No algoritmo da Figura 49 nas linhas 8-13, cada medida de análise é processada sobre cada VA contido no setVA. Com isso, cada VA altera os valores das propriedades de VAarg na linha 12, realizando o processamento das medidas de análise, como detalhado na Figura 50. Após o processamento de todos os VA, algumas medidas precisam ser ajustadas, como a medida AVG. Para isso, definimos um método de correção que é chamado na linha 14 da Figura 49 e detalhado na Figura 51. No final desse processamento, todas as medidas de análise foram processadas e os valores armazenados nas propriedades de VAarg.

```
AgrCA(ConfMesure, hash)
Input: ConfMeasure, hash
Output: graphAnswer
1       graphAnswer ← new  MRGM();
2       for each entry<k,v> in map.entrySet()
3               vertexV ← new VAarg();
4               Set kVD ← entry.getKey();
5               subGraphAgr ← new SubGarg();
6               graphAnswer.insertVD(kVD);
7               graphAnswer.insertVAagr(vertexV);
8               for each va in entry. getValue()
9                       vertexV.put(va)
10                      subGraphAgr.aggregateSubGraph(va.getSubGraph());
11                      for each med in ConfMeasure
12                              vertexV.processingMeasure(med, va);
12                      end for;
13              end for;
14              vertexV.correctMeasure(ConfMeasure);
15              graphAnswer.insertAvd(kVD,vertexV);
16              graphAnswer.insertAva(vertexV.outRelan());
17              graphAnswer.insertSubGagr(subGraphAgr);
18              graphAnswer.insertAsub(vertexV, subGraphAgr);
19      end for;
20      retunr graphAnswer
21      end AgrCA;
```

Figura 49 - Algoritmo de Agregação dos Componentes Analíticos

```
vertexV.processingMeasure(med, va);
Input: med, va
Object: vertexV
1       measure ← med.measure
2       propertyName ← med.property
3       switch(measure)
4               case SUM:
5                       if(this.hasProperty(measure))
6                               value ← this.getProperty(measure);
7                       end if;
8                       this.setProperty(value + va.getProperty(propertyName))
9               end case;
10              case AVG:
11                      if(this.hasProperty(measure))
12                              value[ ] ← this.getProperty(measure);
13                      else
14                              value[ ] ← new number[2];
15                      end if;
16                      value[0] ← value[0] + va.getProperty(propertyName)
17                      value[1] ← value[1] + 1;
18                      this.setProperty(value);
19              end case;
20              case COUNT:
21                      if(this.hasProperty(measure))
22                              value ← this.getProperty(measure);
23                      end if;
24                      this.setProperty(value + 1)
25              end case;
26              case COLLECT:
27                      if(this.hasProperty(measure))
28                              list value ← this.getProperty(measure);
29                      else
30                              value ← new list();
31                      end if;
32                      value.add(va.getProperty(propertyName))
33                      this.setProperty(value);
34              end case;
35              case MIN:
36                      if(this.hasProperty(measure)){
37                              value ← this.getProperty(measure);
38                      else
39                              this.setProperty(va.getProperty(propertyName))
40                      end if;
41                      if(value > va.getProperty(propertyName))
42                              this.setProperty(va.getProperty(propertyName))
43                      end if;
44              end case;
...
```

```
45              case MAX:
46                      if(this.hasProperty(measure)){
47                              value ← this.getProperty(measure);
48                      else
49                              this.setProperty(va.getProperty(propertyName))
50                      end if;
51                      if(value < va.getProperty(propertyName))
52                              this.setProperty(va.getProperty(propertyName))
53                      end if;
54              end case;
55      end switch;
```

Figura 50 - Algoritmo de processamento das medidas de análise

```
vertexV.correctingMeasure(ConfMeasure);
Input: ConfMeasure
Object: vertexV
1       for each med in ConfMeasure
2               measure ← med.measure
3               propertyName ← med.property
4               switch(measure)
5                       case AVG:
6                               if(this.hasProperty(measure))
7                                       value[] ← this.getProperty(measure);
8                                       this.setProperty(value[0]/value[1]);
9                               end if;
10                      end case;
11              end switch;
12      end for;
```

Figura 51 - Algoritmo de correção das medidas de análise

Para a agregação dos Subgrafos Analíticos SubG, é definida um método para agregar os subgrafos analíticos de cada VA do setVA em um subgrafo agregado SubGagr. O método, chamado na linha 10 da Figura 49, recebe os vértices SubV do SubG que se relaciona com o VA. Os vértices SubV recebidos e as arestas SubA, que relacionam esses SubV, são agregados na formação do Subgrafo Agregado SubGagr.

Esses vértices SubV dos SubG constituem um recorte do grafo de dados, de modo que todos os vértices SubV e as arestas SubA representem o conteúdo do grafo de dados. Dessa forma, a produção de um subgrafo agregado SubGagr = (SubVagr, SubAagr) é composta por conjuntos de subgrafos analíticos, representando a agregação dos dados em grafo. Nessa agregação, cada SubVagr representa um conjunto de SubV com determinado rótulo e cada SubAagr representa um conjunto de SubA que possui o mesmo rótulo e relaciona o mesmo par ordenado de vértice. Além disso, na formação do SubGagr, cada

SubVagr estabelece um relacionamento com peso para informar quantas vezes um SubV participou da sua agregação. Com isso, definimos nos algoritmos que a rotulação dos elementos é crucial no processo de agregação de subgrafo, como mostra o algoritmo da Figura 52.

```
subGraphAgr.aggregateSubGraph(va.getSubGraph());
Input: SubG
Object: SubGarg
1       for each vertex in SubG.getVertex()
2               if (mapVertex.containsKey(vertex.Labels))
3                       vetexAgr ← mapVertex.get(vertex.Labels);
4                       vetexAgr.countProperty++;
5               else
6                       vetexAgr ← this.createVertex(vertex.Labels);
7               end if;
8               this.addRelationshipAgd(vetexAgr,vertex);
9       end for;
10      for each edge in SubG.getEdge()
11              if (mapEdge.containsKey(edge.V[0].Labels+edge.Label+edge.V[1].Labels))
12                  edgeAgr ← mapEdge.get(edge.V[0].Labels+edge.Label+edge.V[1].Labels)
13                  edgeAgr.countProperty++;
14              else
15                  vetexAgrSrc ← mapVertex.get(edge.V[0].Labels);
16                  vetexAgrTrc ← mapVertex.get(edge.V[1].Labels);
17              this.createEdgeAgr(vetexAgrSrc.Labels+edge.Label+vetexAgrTrc.Labels);
18              end if;
19      end for;
```

Figura 52 - Algoritmo de agregação dos Subgrafos Analíticos

Na finalização da etapa de Agregação e Visualização, o algoritmo da Figura 49 estrutura os dados de resposta para atender ao MRGM, seguindo a seguinte estrutura:

- cada VD se relaciona com os VAagr, representando suas características;
- cada VAagr se relaciona entre si e com os SubGagr;
- cada vértice do SubGagr se relaciona com vértices V constituinte do grafo de dados.

Tendo em vista o processamento de consulta, concluímos o modelo de CMPGrafo detalhando cada etapa de processamento e os parâmetros de configuração. Para finalizar esse capítulo, apresentamos na próxima seção exemplos de consultas na AAMPGrafo.

5.6 EXEMPLOS DE CONSULTAS NA AAMPGRAFO

Após a especificação da AAMPGrafo, apresentamos a seguir exemplos de consultas multidimensionais com parâmetros de configuração. Nesses exemplos, cada consulta

apresenta quatro parâmetros de configuração e duas ilustrações. Os parâmetros “Consulta do SGBDG” e “Configuração dimensional” são responsáveis por especificar quais dados do SGBDG serão recuperados para a realização da análise e agregação dos dados. A recuperação desses dados forma uma ilustração que retrata o resultado da consulta sobre a modelagem, mostrando os dados a serem agregados.

Já os parâmetros “Configuração topológica” e “Configuração de medida” são responsáveis por especificar a forma de analisar os dados recuperados. Com a realização da análise, mostramos na segunda ilustração os resultados das medidas de análise e a representação visual do modelo de resposta em grafo. Na produção dos exemplos, reutilizamos as modelagens que foram apresentadas na etapa de Seleção.

5.6.1 Exemplos de Consulta na modelagem MPGrafo-1

Relembrando a modelagem MPGrafo-1 na Figura 53, destacamos que a tabela de propriedades dos Vértices Analíticos contém valores pré-processados que podem ser utilizados no cálculo de Medidas de Propriedades (MP) e Medidas Híbridas MH.
Na tabela de propriedade da Figura 53, esses valores condizem com a estrutura topológica da modelagem mostrada, não sendo utilizados em Medidas Topológicas (MT), as quais precisam ser processadas durante a consulta sobre a estrutura topológica dos dados recuperados. Com isso, apresentamos a seguir os exemplos de consulta na modelagem MPGrafo-1.

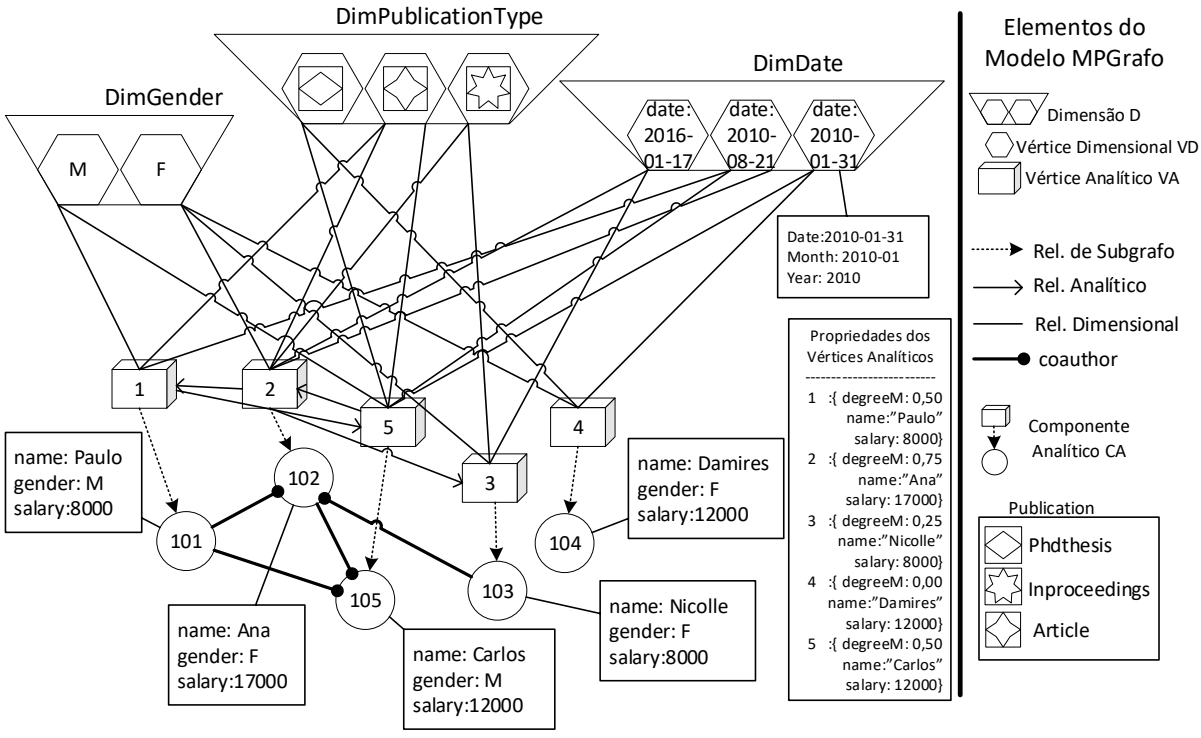


Figura 53 - Modelagem MPGrafo-1 com apenas um vértice :Person no subgrafo

❖ Na exemplificação de consultas, apresentamos na Figura 54 o resultado da consulta com os seguintes parâmetros:

- Consulta do SGBDG: consultar os autores que publicaram;

  $\bowtie_{src_1,\ trg_1}^{\phi E_{VA}} \left(\bowtie_{src_2,\ trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

  - ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge\ \lambda\ (trg_1) = r_{VA}$
  - ϕE_{sub}: $trg_1 = src_2 \wedge src_2 = trg_2 \wedge \lambda\ (src_2) = r_{VA}$
- Configuração dimensional: {DimDate:[year]}
- Configuração topológica: { *betweenness*:[collect]}
- Configuração de medida: { name:[collect], salary:[avg], degreeM:[min] }

Com os parâmetros “Consulta do SGBDG” e “Configuração dimensional”, recuperamos os dados e unificamos os vértices dimensionais, considerando o rótulo “DimDate” e os valores existentes na propriedade “year”. Os dados recuperados são ilustrados na Figura 54A.

Com o parâmetro “Configuração topológica” é realizada a execução do algoritmo *betweenness* para computar a centralidade de cada VA recuperado. Em seguida, na etapa de agregação e visualização, é realizada a agregação dos Componentes Analíticos, processando as medidas de análise requisitada nos parâmetros.

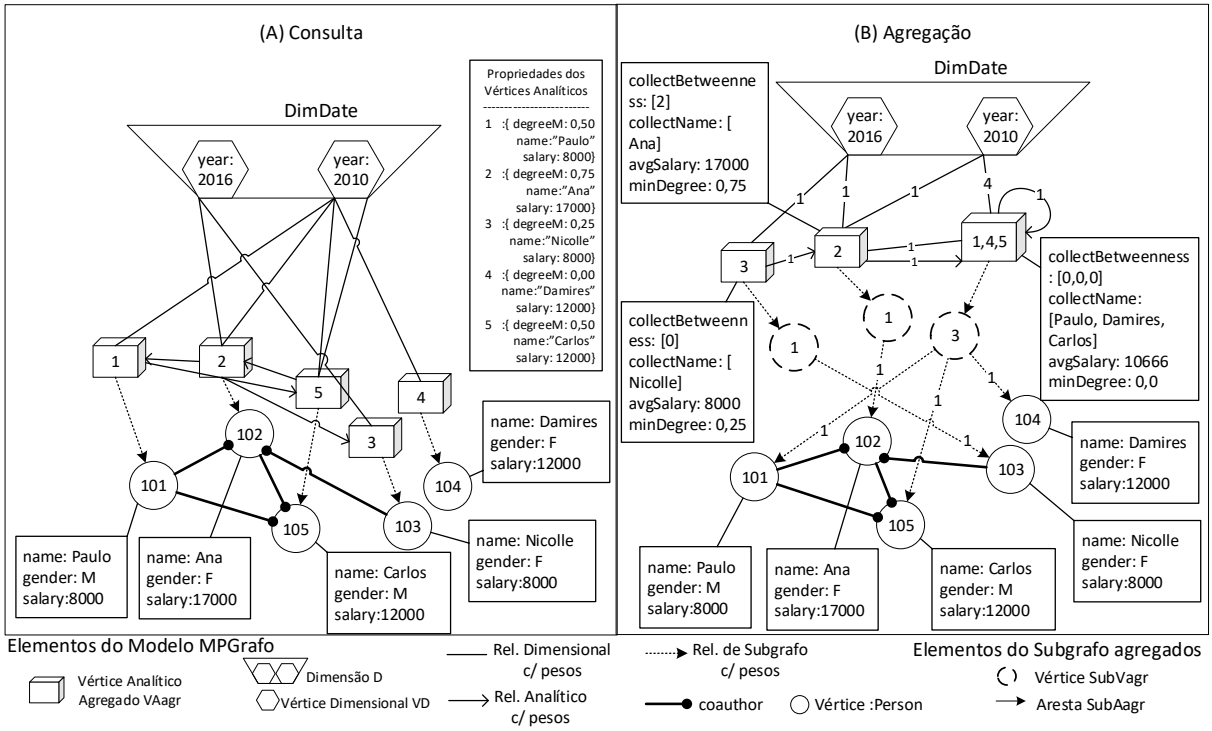


Figura 54 - Consulta os autores que publicaram e os agrega por ano

A Figura 54B apresenta um modelo de resposta contendo: dois vértices dimensionais,

três VAagr com as identificações dos VA integrantes e três SubGagr, os quais são relacionados com os vértices do grafo de dados.

Nas propriedades dos VAagr da Figura 54 constam as seguintes medidas de análise: collectBetweenness, que informa uma lista de dados contendo os valores da centralidade betweenness de cada VA que constitui o VAagr; collectName, que informa um lista de dados contendo os valores da propriedade “name”; avgSalary, que calcula a média dos valores da propriedade “salary”; e minDegreeM, que informa o menor valor da propriedade “degreeM”.

❖ Na Figura 55 apresentamos a resposta da consulta que possui os seguintes parâmetros:

- Consulta do SGBDG: consultar os autores que publicaram em 2010;

  $\bowtie_{src_1,\ trg_1}^{\phi E_{VA}} \left(\bowtie_{src_2,\ trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

  - ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate ∧ src_1.year = 2010 ∧ $\lambda\ (trg_1) = r_{VA}$
  - ϕE_{sub}: $trg_1 = src_2 \wedge src_2 = trg_2 \wedge \lambda\ (src_2) = r_{VA}$
- Configuração dimensional: {DimDate:[year]}
- Configuração topológica: { degree:[sum, avg] , *closeness*:[sum] }
- Configuração de medida: { name:[collect], salary:[sum], degreeM:[sum, avg] }

Com os parâmetros “Consulta do SGBDG” e “Configuração dimensional”, recuperamos o VD da dimensão DimDate que possue o valor 2010 na propriedade “year”, como mostra a Figura 55A.

O parâmetro “Configuração topológica” requisita o processamento dos algoritmos degree e *closeness* para computar o valor de centralidade de cada algoritmo nos VA recuperado. Em seguida, na etapa de Agregação e Visualização, é realizada a agregação dos Componentes Analíticos, processando as medidas de análise requisitadas nos parâmetros.

A Figura 55B apresenta um modelo de resposta contendo: um vértice dimensional, um VAagr com as identificações dos VA integrantes e um SubGagr, que é relacionado com os vértices do grafo de dados. Além disso, essa consulta equivale à operação de “Slice” em relação à consulta da Figura 55, de modo que fatia os dados da consulta especificando o valor “2010” na propriedade “year” da dimensão “DimDate”.

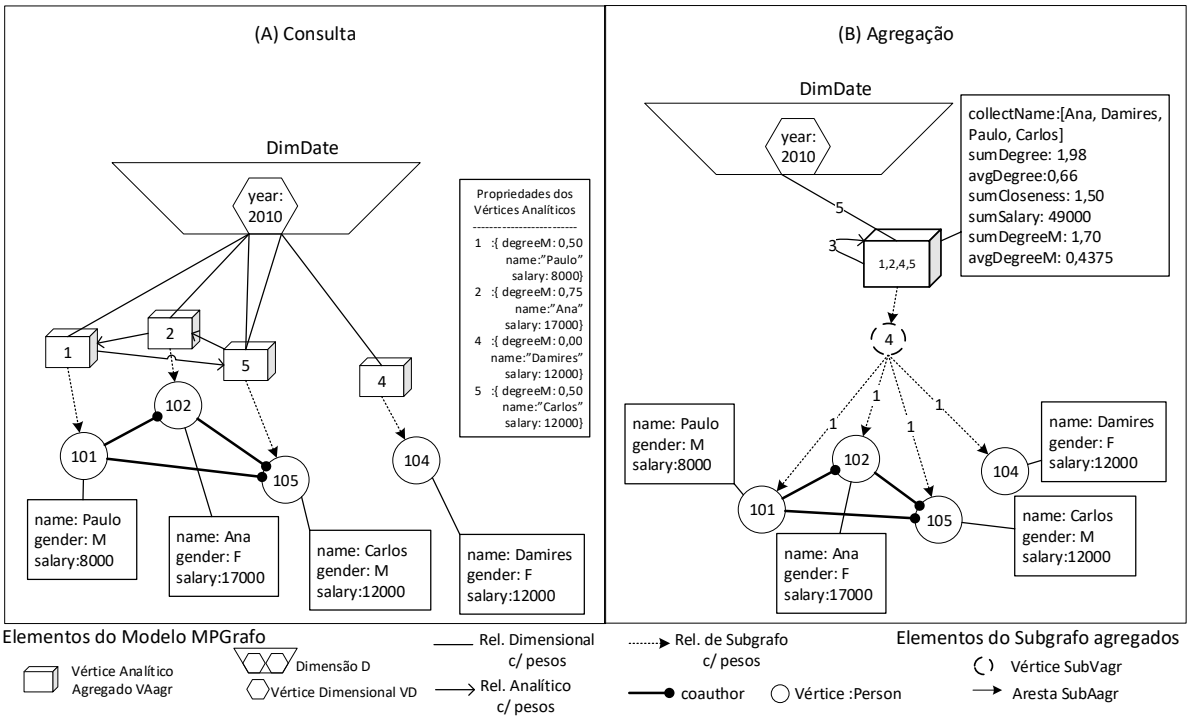


Figura 55 - Consulta e agrega os autores que publicaram em 2010

Nas propriedades dos VAagr da Figura 55 constam as seguintes medidas de análise: collectName, que informa um lista de dados contendo os valores da propriedade “name”; sumDegree, que soma o grau de centralidade dos VA na consulta; avgDegree, que calcula a média do grau de centralidade na consulta; sumCloseness, que processa o algoritmo closeness e soma os valores dos VA na consulta; sumSalary, que soma os valores da propriedade “salary”; sumDegreeM, que soma os valores da propriedade “degreeM”; e avgDegreeM, que calcula a média dos valores na propriedade “degreeM”.

❖ Na Figura 56 apresentamos a resposta da consulta que possui os seguintes parâmetros:

- Consulta do SGBDG: selecionar os autores;

  $\bowtie_{src_1,\ trg_1}^{\phi E_{VA}} \left(\bowtie_{src_2,\ trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

  - ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge\ \lambda\ (trg_1) = r_{VA}$
  - ϕE_{sub}: $trg_1 = src_2$
- Configuração dimensional: {DimDate:[month]}
- Configuração topológica: { *closeness*:[avg] }
- Configuração de medida: { *:count, degreeM:[sum]}

Com os parâmetros “Consulta do SGBDG” e “Configuração dimensional”, recuperamos os VD da dimensão DimDate com distintos valores na propriedade “month”, como mostra a Figura 56A. O parâmetro “Configuração topológica” requisita o processamento do algoritmo

closeness para computar o valor de centralidade dos VA recuperado.

A Figura 56B apresenta um modelo de resposta contendo: três vértices dimensionais, três VAagr com as identificações dos VA integrantes e três SubGagr, o quais são relacionados com os vértices do grafo de dados. Além disso, essa consulta equivale à operação de “Drill-down” na dimensão “DimDate” em relação à consulta da Figura 55, da mesma forma que a Figura 55 representa a operação contrária “Roll-up” em relação a consulta da Figura 56. Com isso, observamos as diferenças dos resultados de agregação em função das operações OLAP.

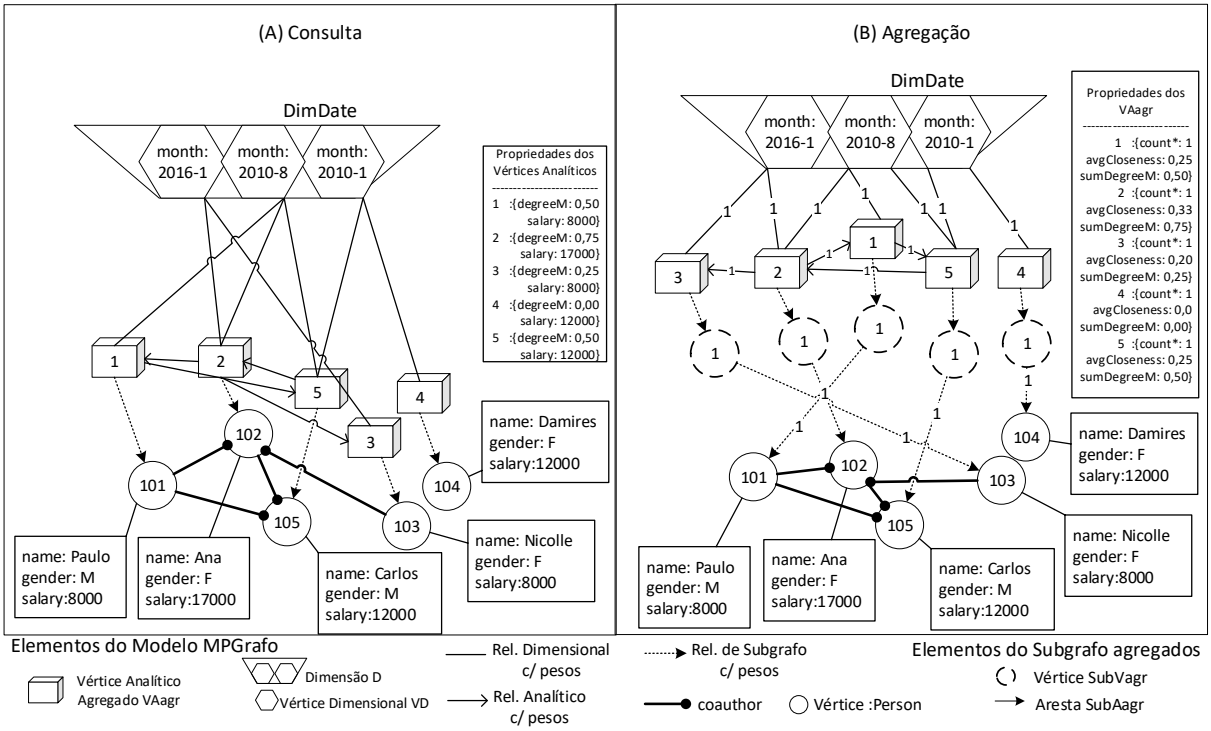


Figura 56 - Consulta os autores que publicaram e os agrega pelos meses

Nas propriedades dos VAagr da Figura 56 constam as seguintes medidas de análise: count*, que informa a quantidade de VA integrante no VAagr; sumDegreeM, que a soma dos valores da propriedade “degreeM”; avgCloseness, que processa o algoritmo closeness e calcula a média dos valores dos VA.

❖ Na Figura 57 apresentamos a resposta da consulta que possui os seguintes parâmetros:

- Consulta do SGBDG: consultar os autores do gênero feminino que publicaram;

$\bowtie^{\phi E_{VA}}_{src_1,trg_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{src_3,\ trg_3} (r_{relsub})\right)$, onde

  - ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge\ \lambda\,(trg_1) = r_{VA} \wedge \lambda\ (src_2)$ = :DimGender $\wedge$ src_2.gender = “F” $\wedge\ \lambda\ (trg_2) = r_{VA}$
  - ϕE_{sub}: $trg_1 = src_3 \wedge\ trg_2 = src_3 \wedge src_3 = trg_3 \wedge \lambda\ (src_3) = r_{VA}$

- Configuração dimensional: {DimDate:[year]}
- Configuração topológica: { }
- Configuração de medida: { *:count, salary:[avg], degreeM:[avg]}

Com os parâmetros “Consulta do SGBDG” e “Configuração dimensional”, recuperamos VD da dimensão “DimDate” com diferentes valores na propriedade “year”, como mostra a Figura 57A. Nessa consulta, medidas topológicas não são requisitadas no parâmetro “Configuração topológica”.

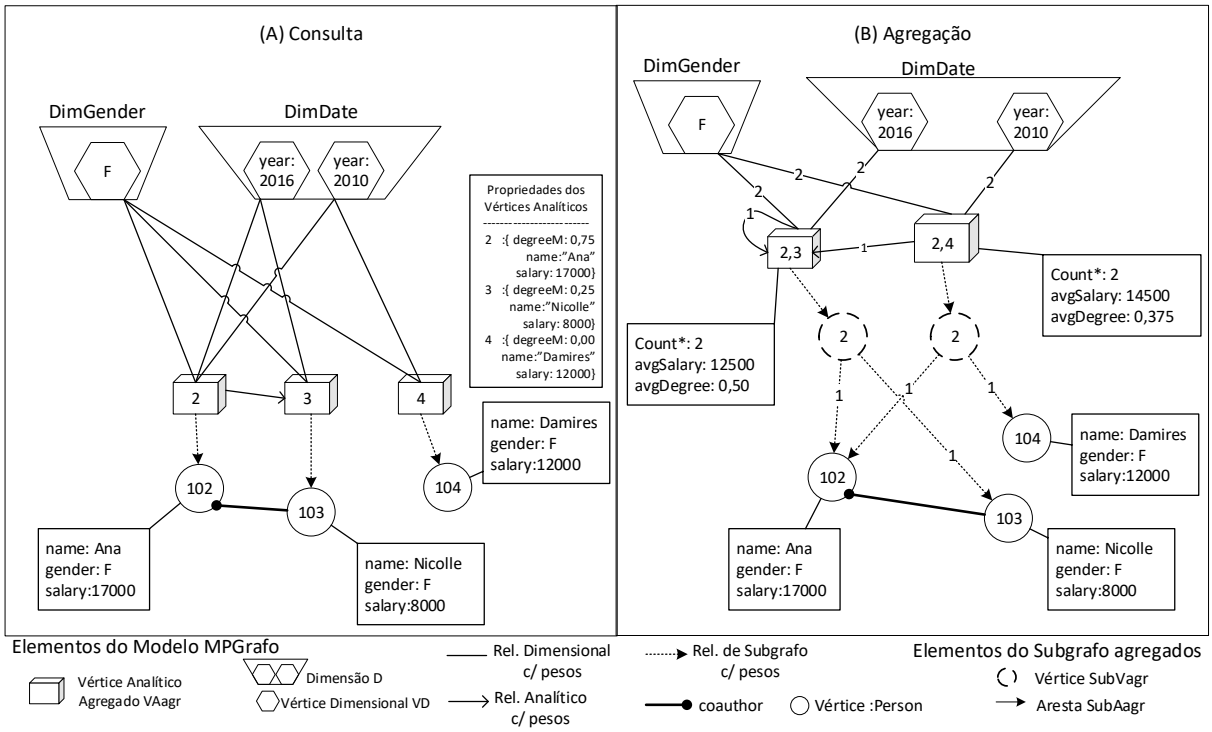


Figura 57 - Consulta os autores do gênero feminino que publicaram e os agrega por ano

A Figura 57B apresenta o modelo de resposta contendo: três vértices dimensionais, dois VAagr com as identificações dos VA integrantes e dois SubGagr. Nessa consulta, duas dimensões diferentes são utilizadas para selecionar os dados, mostrando o uso de múltiplas dimensões nas consultas. A Figura 57 mostra as seguintes medidas de análise: count*, que informa a quantidade de VA integrante no VAagr; avgSalary, que calcula a média dos valores da propriedade “salary”; avgDegreeM, que calcula a média dos valores da propriedade “degreeM”.

5.6.2 Exemplos de Consulta na modelagem MPGrafo-2

Relembrando a modelagem MPGrafo-2 na Figura 58, destacamos que a tabela de propriedades dos Vértices Analíticos contém valores pré-processados que podem ser utilizados no cálculo de Medidas de Propriedades (MP). Essa modelagem não possui grafo

de fatos entre os VA, portanto, não computa Medidas Topológicas (MT) ou Medidas Híbridas (MH). Assim, apresentamos a seguir os exemplos de consulta na modelagem MPGrafo-2.

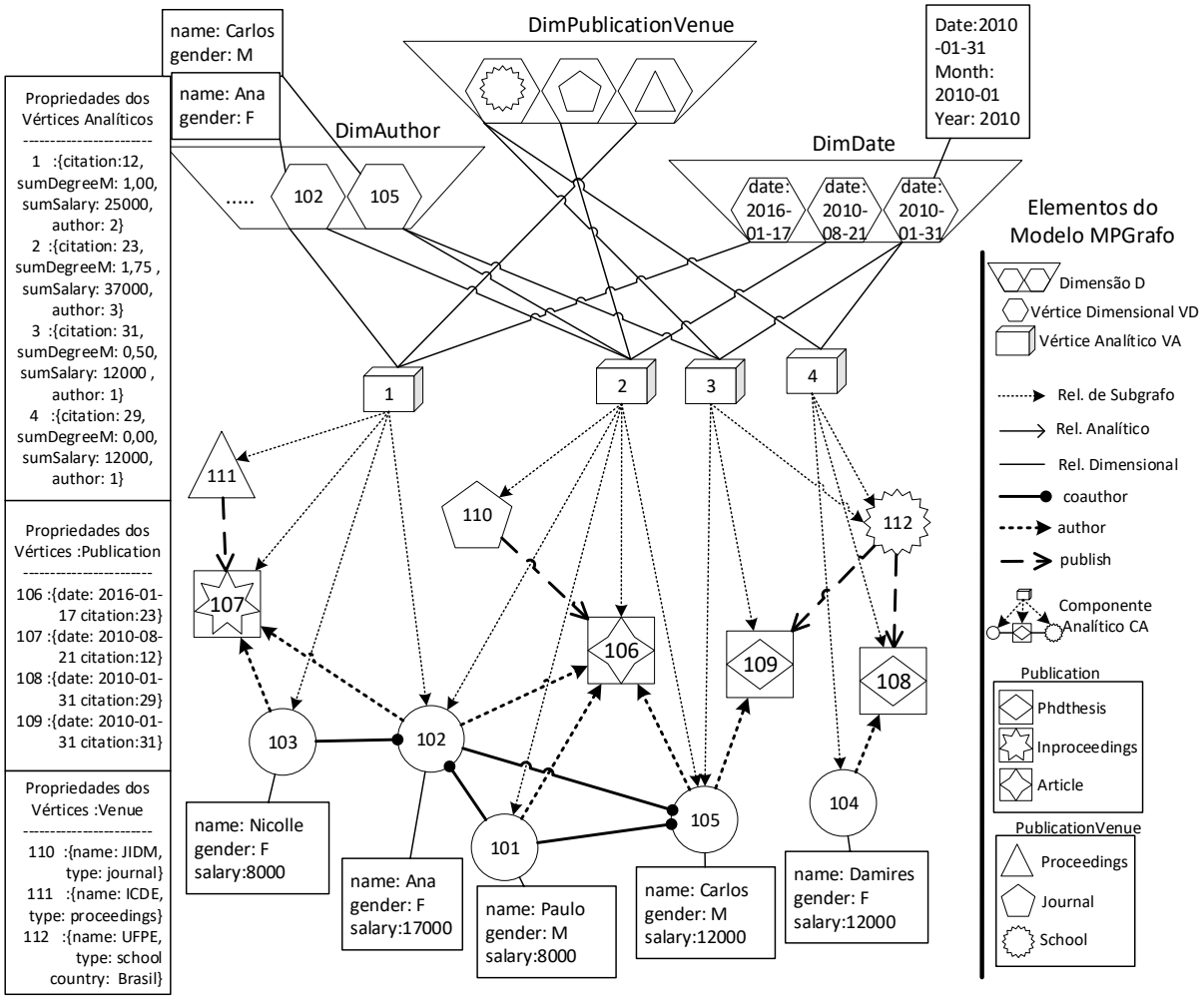


Figura 58 - Modelagem MPGrafo-2 com um conjunto de vértices no subgrafo

- No exemplo de consulta das Figuras 59 e 60, mostramos o resultado da consulta que possui os seguintes parâmetros:
  - Consulta do SGBDG: consultar as publicações considerando data e local;

    $\bowtie^{\phi E_{VA}}_{src_1,trg_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{trg_3} (r_{relsub})\right)$, onde
    - ϕE_{VA}: $\lambda\ (src_1)$ = ":DimDate" $\wedge\ \lambda\,(trg_1) = r_{VA} \wedge \lambda\ (src_2)$ = ":DimPublicationVenue" $\wedge\ \lambda\ (trg_2) = r_{VA}$
    - ϕE_{sub}: $trg_1 = src_3 \wedge trg_2 = src_3$
  - Configuração dimensional: {DimDate:[year], DimPublicationVenue:[type]}
  - Configuração topológica: { }
  - Configuração de medida: { *:count, citation:[sum], author:[sum]}

Com os parâmetros “Consulta do SGBDG” e “Configuração dimensional”, recuperamos os dados e unificamos os vértices dimensionais de duas dimensões, considerando na dimensão “DimDate” os valores existentes na propriedade “year” e na dimensão “DimPublicationVenue” os valores de “type”, como mostra a Figura 59.

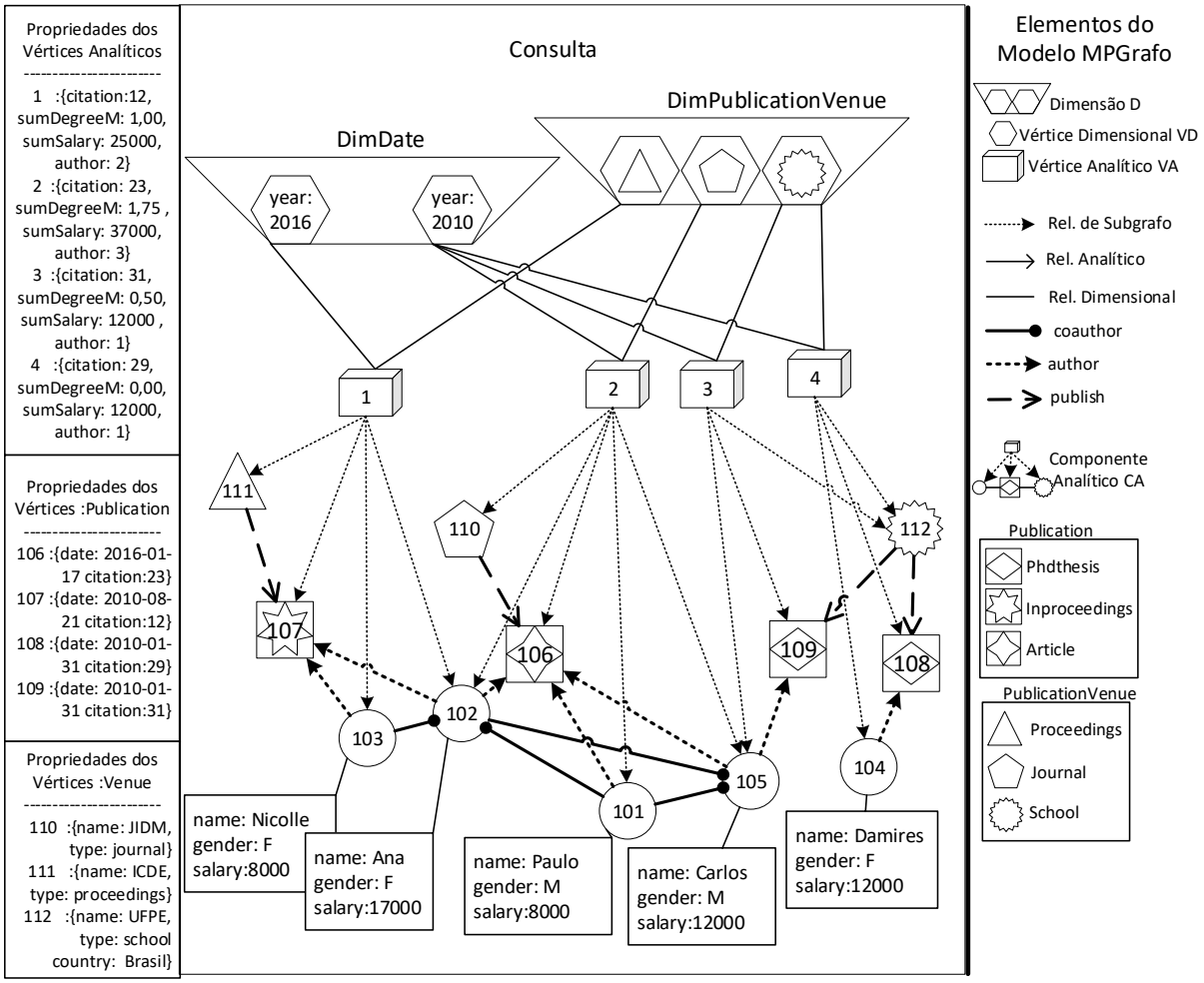


Figura 59 - Consulta as publicações considerando ano e tipo de local

A Figura 60 apresenta o modelo de resposta resultante, contendo: cinco vértices dimensionais, três VAagr com as identificações dos VA integrantes e três SubGagr. No SubGagr, cada elemento representa um rótulo, agregando assim todos os elementos do Subgrafo Analítico SubG que possuem o mesmo rótulo. Nas ilustrações com grafo de resposta, os SubGagr são destacados em uma área com linhas tracejadas.

Nesse exemplo de consulta, duas dimensões diferentes são utilizadas para selecionar os dados, mostrando a aplicação de múltiplas dimensões em modelagens que analisam padrões em grafo. Na Figura 60 são apresentadas as seguintes medidas de análise: count*, que informa a quantidade de VA integrante no VAagr; sumCitation, que soma os valores da propriedade “citation”; sumAuthor, que soma os valores da propriedade “author”.

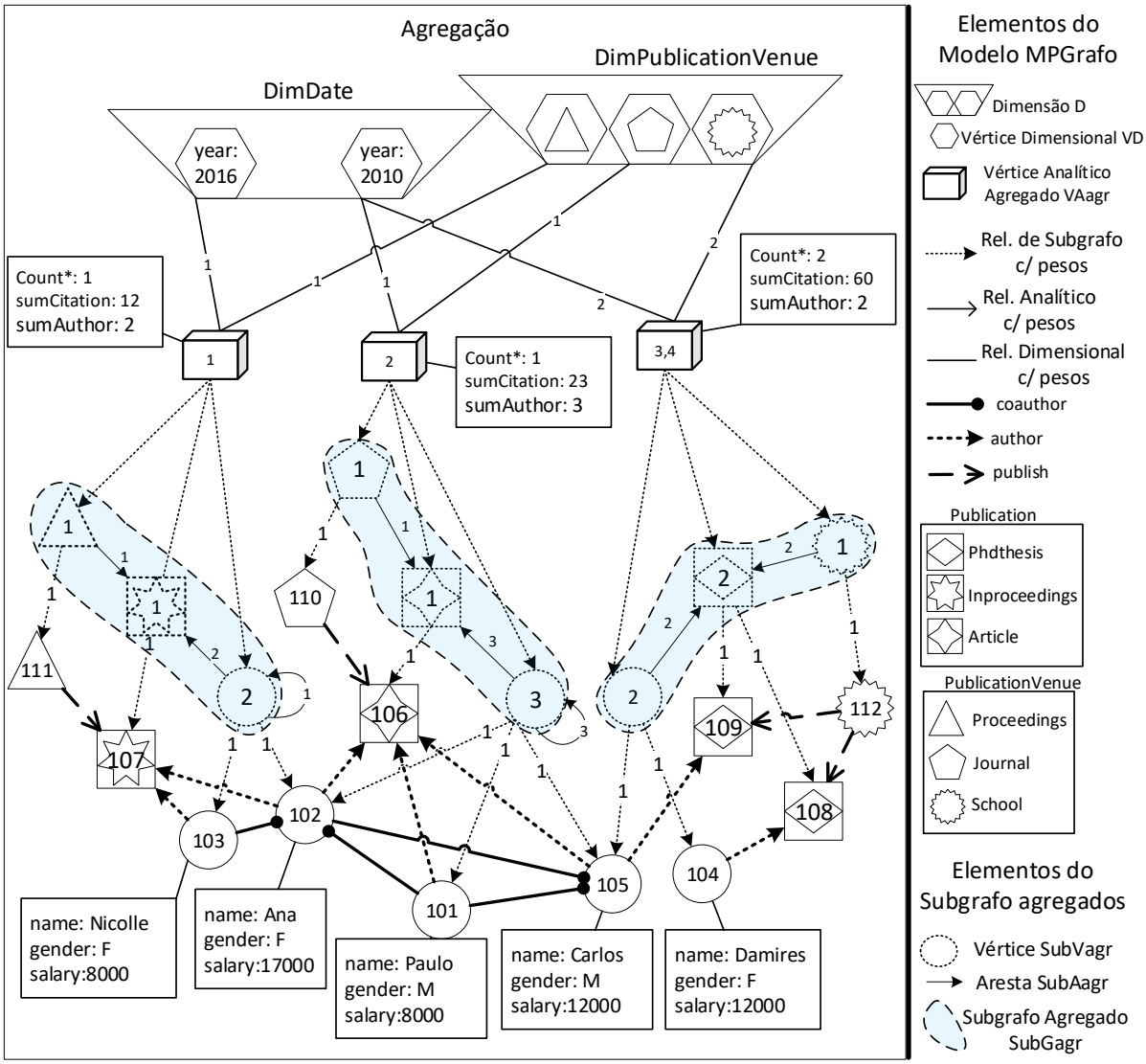


Figura 60 - Consulta e agrega as informações de publicação considerando ano e tipo de local

- No exemplo de consulta das Figuras 61 e 62, mostramos o resultado da consulta que possui os seguintes parâmetros:
  - Consulta do SGBDG: consultar as publicações considerando data e local ;

    $\bowtie^{\phi E_{VA}}_{src_1,trg_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{trg_3} (r_{relsub})\right)$, onde
    - ϕE_{VA}: $\lambda\ (src_1)$ = ":DimDate" $\wedge\ src_1$.year = 2010 $\wedge\ \lambda\ (trg_1) = r_{VA} \wedge \lambda\ (src_2)$ = ":DimPublicationVenue" $\wedge\ src_2$.type = "School" $\wedge\ \lambda\ (trg_2) = r_{VA}$
    - ϕE_{sub}: $trg_1 = src_3 \wedge\ trg_2 = src_3$
  - Configuração dimensional: {DimDate:[year], DimPublicationVenue:[type]}
  - Configuração topológica: { }
  - Configuração de medida: { *:count, citation:[sum], author:[sum]}

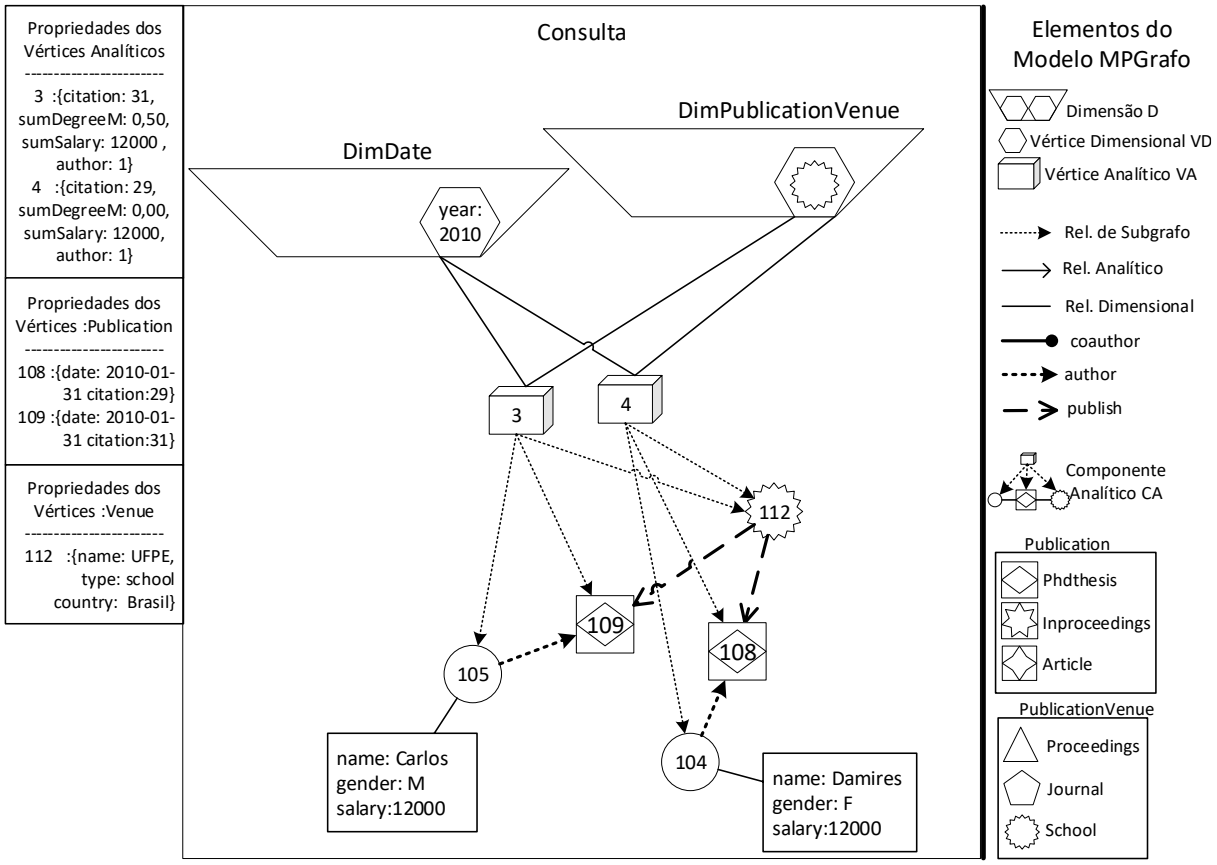


Figura 61 - Consulta as publicações de 2010 em instituições

Com os parâmetros “Consulta do SGBDG” e “Configuração dimensional”, recuperamos os VD da dimensão “DimDate” com valor 2010 na propriedade “year” e da dimensão “DimPublicationVenue” com valor “School” na “type”. Essa consulta corresponde à operação “Dice” sobre a consulta da Figura 60, ou seja, a visualização da dimensão “DimDate” e “DimPublicationVenue” são restringidas pelos valores 2010 na propriedade “year” e “School” na propriedade “type”, respectivamente, recuperando os dados da Figura 62.

A Figura 62 apresenta o modelo de resposta resultante, contendo: dois vértices dimensionais, um VAagr com as identificações dos VA integrantes e um SubGagr.

Nesse exemplo de consulta a Figura 62 mostra as mesmas medidas de análise da Figura 60.

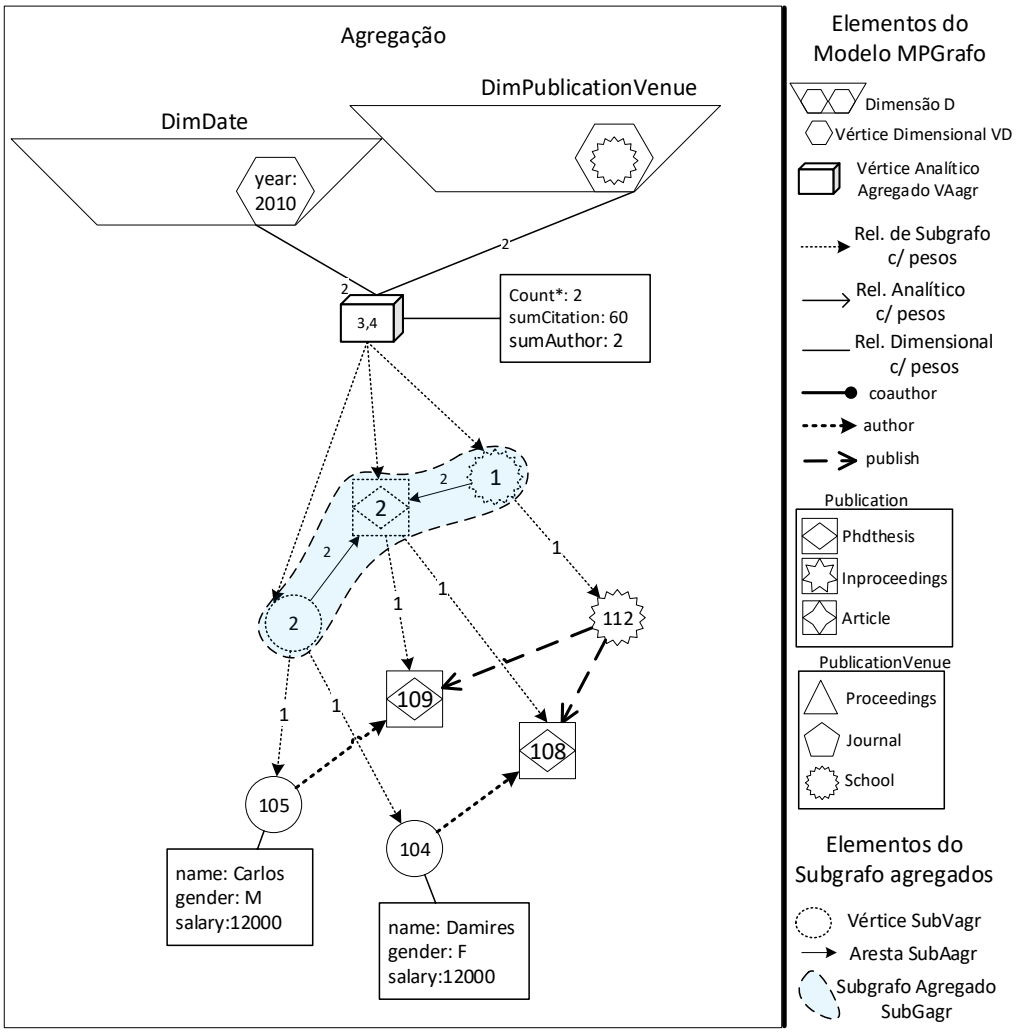


Figura 62 - Consulta agregada das publicações de 2010 em instituições

Por fim, com a representação dos exemplos de consulta no modelo de consulta em grafo pudemos mostrar consultas sem associar a uma linguagem de consulta em grafo proprietária de um SGBDG, de modo que essa abordagem poderia ser desenvolvida em qualquer SGBDG que atenda ao modelo de grafo de propriedade.

5.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, foi abordado o modelo de Consulta Multidimensional em Padrões de Grafo (CMPGrafo), detalhando os parâmetros e as etapas de processamento da consulta. Os parâmetros da consulta são definidos e detalhados para especificar e combinar os recursos de análise na execução da consulta. As etapas de processamento são detalhadas e explanadas junto aos parâmetros da consulta. A etapa de Seleção mostra por meio do MCG exemplos de consultas em SGBDG. A etapa de Análise Topológica especifica os algoritmos de análise de rede incorporados na abordagem. A etapa de Agregação e

Visualização apresenta os algoritmos utilizados na agregação de dados em grafo e o modelo de resposta em grafo multidimensional. Por fim, mostramos representações de consultas multidimensionais em padrões de grafos, aplicando operadores OLAP e medidas de análise. No próximo capítulo apresentamos a implementação de uma instância dessa abordagem de análise no SGBDG Neo4j.

6 IMPLEMENTAÇÃO DA AAMPGRAFO

Instanciamos nossa AAMPGrafo sobre o SGBDG Neo4j, o qual atende ao Modelo de Grafo de Propriedade e utiliza a linguagem de consulta Cypher[16] para manipular os dados. Na implementação da abordagem, desenvolvemos um *plugin* para incorporar nas consultas recursos para a análise topológica e analítica do grafo de dados. Esse *plugin* utiliza a interface gráfica do Neo4j para reproduzir os resultados das consultas. Além disso, apresentamos nesse capítulo o funcionamento da AAMPGrafo reproduzindo as modelagens e as consultas utilizadas nos capítulos anteriores.

Este capítulo cobre o seguinte conteúdo. A Seção 6.1 detalha os recursos utilizados na implementação e especifica os parâmetros utilizados nas consultas. A Seção 6.2 apresenta a utilização da AAMPGrafo implementada, mostrando exemplos de consultas e a representação gráfica das respostas. Na Seção 6.3 discutimos sobre a representação visual das respostas das consultas na literatura. Na Seção 6.4 realizamos uma análise comparativa entre as características dos trabalhos relacionados e a AAMPGrafo que desenvolvemos neste trabalho. Por fim, a Seção 6.5 conclui o capítulo com as considerações finais.

6.1 IMPLEMENTAÇÃO

Neste trabalho, desenvolvemos um *plugin* que incorpora no SGBDG as tecnologias OLAP e os recursos de análise em grafo. Na escolha do SGBDG, consideramos que o mesmo precisaria satisfazer oas seguintes requisitos: (i) atender ao modelo de grafo de propriedade, visto que é o modelo mais difundido no mercado (BONIFATI et al., 2018); (ii) ser um SGBDG nativo para não comprometer ao desempenho na travessia dos relacionamentos, pois a estrutura em grafo de SGBDG nativo é mantida tanto no armazenamento físico quanto no processamento das consultas; e (iii) ter o código aberto para permitir o desenvolvimento de um *plugin*. Escolhemos então o SGBDG Neo4j para implementar o *plugin*, visto que atende todos os requisitos e ainda disponibiliza uma interface gráfica para visualizar a resposta das consultas em grafo.

Na implementação da AAMPGrafo, definimos o Modelo MPGrafo para estruturar os dados e viabilizar o processamento de consultas multidimensionais no SGBDG e um *plugin*

[16] https://neo4j.com/developer/cypher-query-language/

para incorporar ao Neo4j uma funcionalidade que permita processar consultas multidimensionais em grafo.

O Modelo MPGrafo foi definido para estruturar o grafo de dados de modo independente de SGBDG. Assim, para exemplificar o Modelo MPGrafo apresentamos imagens do Neo4j contendo as modelagens MPGrafo-1 e MPGrafo-2, que foram exibidas no Capítulo 5.

Na implementação do *plugin*, reutilizamos o código fonte de um *plugin* existente, denominado APOC[17] (*A Package Of Component for Neo4j*), para mediar a integração com o Neo4j. Com isso, reusamos a codificação do APOC para desenvolver uma função parametrizada, encarregada de receber os parâmetros de configuração e de realizar o processamento de consultas multidimensionais em grafo.

O *plugin* foi desenvolvido em JAVA para a versão (neo4j-community-3.3.4) de código aberto do Neo4j. A execução da função é realizada via interface gráfica do Neo4j, como se fosse uma extensão dos recursos de análise do próprio SGBDG. O processo de consulta já foi detalhado na descrição do modelo de CMPGrafo, sendo executado no *plugin* dentro do programa Neo4j. Com isso, apresentamos a seguir a utilização do *plugin* mostrando primeiro o formato da função e os parâmetros de configuração.

6.1.1 Parâmetros da consulta

Os parâmetros da consulta consistem em compor um conjunto de informações estruturadas para especificar as formas de análise na consulta multidimensional em grafo. Esses parâmetros definem na função a forma de executar e analisar os dados requeridos na consulta. A chamada da função obedece ao seguinte formato:

```
1   Call olap.aggregate.graphResulta(
2   Query SGBDG,
3   [{ Property or Algorithm : [function,...], ... }, { Property : [function,...] }],
4   [{SubGV:{ Property : [function,...] }, ... }],
5   {r_VD : [property,...] })
```

Figura 63 - Formato da consulta multidimensional em grafo

A Figura 63 mostra a representação da função *"olap.aggregate.graphResulta"* com a especificação dos parâmetros, os quais detalhamos, a seguir:

- Na linha 2 "*Query SGBDG*" é o parâmetro encarregado de receber uma consulta em

[17] https://neo4j-contrib.github.io/neo4j-apoc-procedures/

Cypher que segue o modelo de consulta MCG. No resultado da consulta, é necessário retornar os seguintes dados: os relacionamentos dimensionais (reldi) entre os VD e os VA, os relacionamentos analíticos (relan) entre os VA, os VA e os relacionamentos de subgrafo (relsub) entre os VA e os vértices dos subgrafo analíticos (SubGV). Os relan são facultativos na consulta sendo necessários para o processamento de algoritmos de análise em grafo. Esse parâmetro será detalhado na próxima seção.

- Na linha 3 há um parâmetro com dois mapeamentos de configuração. O primeiro especifica a forma de analisar os VA, definindo um lista no seguinte formato *{ Property or Algorithm : [function,...], ... }*, que requisita o processamento de funções agregadas (*[function,...]*) sobre os valores de propriedades dos VA ou sobre os valores resultantes dos algoritmos. O segundo especifica a forma de analisar os relacionamentos entre os VA, definindo uma lista no formato *{ Property : [function,...] }*, que requisita o processamento de funções agregadas sobre os valores de propriedades do relacionamento. Com isso, cada mapeamento contém uma lista de configurações, que podem ser exemplificadas da seguinte forma:
  - “salary:['sum','avg','min','max']” requisita o processamento dessa lista de funções agregadas sobre os valores da propriedade “salary” do VA;
  - betweenness:['sum','collect'] requisita o processamento dessa lista de funções agregadas sobre os valores resultantes do algoritmo betweenness; e
  - {`*`:'count'} requisita a contagem dos elementos que compõem a agregação.
- Na linha 4 há um parâmetro contendo uma lista de mapeamentos de configuração. Cada mapeamento contém um rótulo do SubGV e uma lista de configuração, que atende ao seguinte formato *{ Property : [function,...] }*. Com isso, é possível definir uma lista de configurações para cada rótulo de vértice que compõe o subgrafo agregado. Esse parâmetro é composto por uma lista de mapeamentos que atendem ao seguinte exemplo:
  - Person:{salary:['sum','avg'],degreeM:['sum','avg']}, onde “Person” é o rótulo do SubGV e {salary:['sum','avg'],degreeM:['sum','avg']} é uma lista de configurações que requisita o processamento das funções de agregação “:['sum','avg']” sobre os valores da propriedade “salary” do VA.
- Na linha 5 há um parâmetro que determina a configuração dimensional informando uma lista contendo em cada elemento, um rótulo da dimensão r_{VD} e um conjunto de

propriedades dessa dimensão. Por exemplo:

- {DimDate:'year'}, informa a dimensão “DimDate” e a propriedade “year” dessa dimensão.

Após a descrição dos parâmetros, detalharemos o parâmetro *Query SGBDG,* mostrando a correspondência entre o MCG (modelo de consulta) e a linguagem de consulta Cypher.

6.1.2 Modelo de Consulta na Linguagem Cypher

Definimos neste trabalho, um modelo de consulta baseado em RPGA para especificar consultas em grafo que independam de SGBDG, podendo adequar o modelo MCG para outras liguagens de consultas em SGBDG. Com a implementação da AAMPGrafo no Neo4j, apresentaremos as consultas do MCG na linguagem Cypher mostrando uma correspondência entre o MCG e a linguagem referida.

A Figura 64 apresenta duas representações de consultas uma em MCG, que já foi abordada na Figura 37, e a outra na linguagem Cypher. Essas consultas são similares, solicitando as mesmas informações na modelagem MPGrafo-1. A consulta em MCG requisita os relacionamentos entre src_1 e trg_1, de modo que os vértices src_1 tenham o rótulo “DimGender” e possuam o valor “F” na propriedade “gender” e os vértices trg_1 tenham o rótulo r_{VA}. Com a recuperação desses relacionamentos, são realizadas duas consultas aninhadas. A primeira solicita os relacionamentos, que possuem o rótulo r_{relsub}, e relacionam os vértices src_2 e os vértices trg_2, de modo que os vértices src_2 são os vértices trg_1 da consulta anterior e os vértices trg_2 completam os relacionamentos consultados. A segunda solicita os relacionamentos entre os vértices src_2, ou seja, os relacionamentos que relacionam dois vértices de rótulo r_{VA}.

A representação dessa consulta MCG em Cypher requisita na expressão “(dim:DimGender)-[reldi]->(va: r_{VA})” os relacionamentos “reldi” entre os vértices “dim” de rótulo “DimGender” e os vértices “va” de rótulo r_{VA}. Em seguida, a expressão “(va: r_{VA})-[relsub: r_{relsub}]->(SubGV)” requisita os relacionamentos “relsub” de rótulo r_{relsub} entre os vértices “va” da expressão anterior e os vértices “SubGV” de qualquer rótulo. Na próxima expressão (: r_{VA})-[relan]->(: r_{VA}) são requisitados os relacionamentos “relan” entre os vértices de rótulo r_{VA}.

No final da consulta, os vértices “dim” são restringidos, com a expressão “Where dim.gender = ‘F’ ”, para selecionar apenas os que possuem valor ‘F’ na propriedade “gender”. Por fim,

com a execução da consulta em Cypher são recuperados os relacionamentos “reldi”, “relan” e “relsub” e os vértices “va”.

Consulta no MCG
$\bowtie^{\phi E_{VA}}_{src_1,\ trg_1} \left(\bowtie^{\phi E_{sub}}_{src_2,\ trg_2} (r_{relsub}) \right)$, onde • ϕE_{VA}: $\lambda\ (src_1)$ = :DimGender ∧ src_1.gender = “F” ∧ $\lambda\ (trg_1) = r_{VA}$ • ϕE_{sub}: $trg_1 = src_2 \wedge src_2 = trg_2 \wedge \lambda\ (src_2) = r_{VA}$

Consulta em Cypher	
1	MATCH
2	(dim:DimGender)-[reldi]->(va: r_{VA}),
3	(va: r_{VA})-[relsub: r_{relsub}]->(SubGV),
4	(: r_{VA})-[relan]->(: r_{VA})
5	Where dim.gender = ‘F’
6	RETURN reldi, relan, relsub, va;

Figura 64 - Correspondência entre o modelo de consulta e a linguagem Cypher

Mediante a correspondência entre as formas de consultar os dados no SGBDG, introduzimos a linguagem Cypher nos próximos exemplos seguindo esse formato de consulta. Na próxima seção, mostramos o funcionamento do *plugin* por meio de ilustrações do Neo4j.

6.2 UTILIZAÇÃO DA AAMPGRAFO

Para a utilização da AAMPGrafo, é necessário instalar o *plugin* no Neo4j para incluir a funcionalidade que permite processar consultas multidimensionais em grafo. Em seguida, é necessário modelar o grafo de dados no Neo4j de acordo com o Modelo MPGrafo. Assim, o sistema está pronto para realizar as consultas multidimensionais em grafo. Para se familiarizar com a interface gráfica do Neo4j, apresentamos na Figura 65 a representação visual do grafo de dados da amostra do DBLP. Essa representação equivale à Figura 24, contendo os mesmos dados no Neo4j.

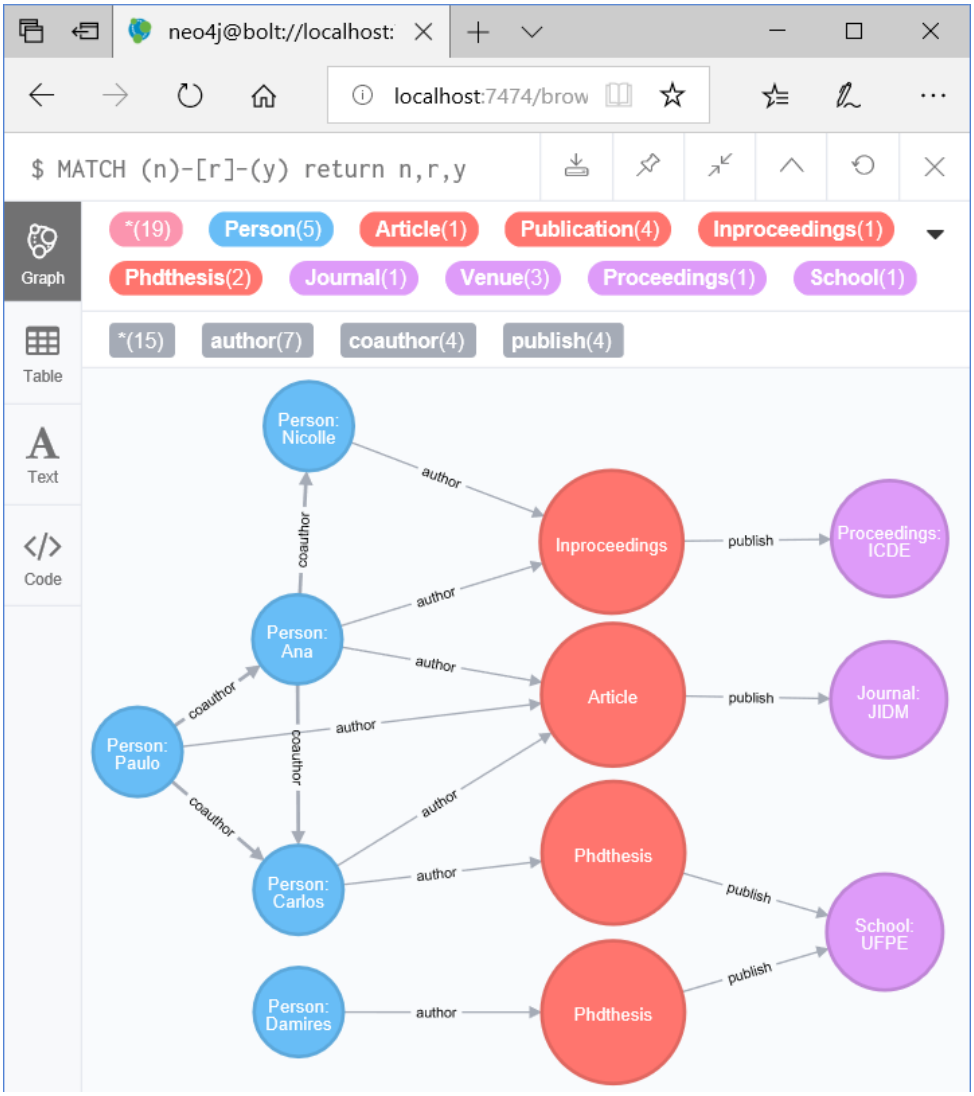


Figura 65 - Amostra DBLP no Neo4j

Na exibição dos dados no Neo4j, é possível ajustar a visualização do grafo, de modo que o tamanho, a cor e a propriedade dos elementos do grafo são ajustáveis em função do rótulo. Por exemplo, a visualização do grafo na Figura 66 foi configurada, de forma que os vértices de rótulo (:Person) apresentem a cor azul e informem o conteúdo da propriedade “nodeName”. Além disso, é possível selecionar na interface um elemento para mostrar seu conteúdo, como mostra a Figura 66.

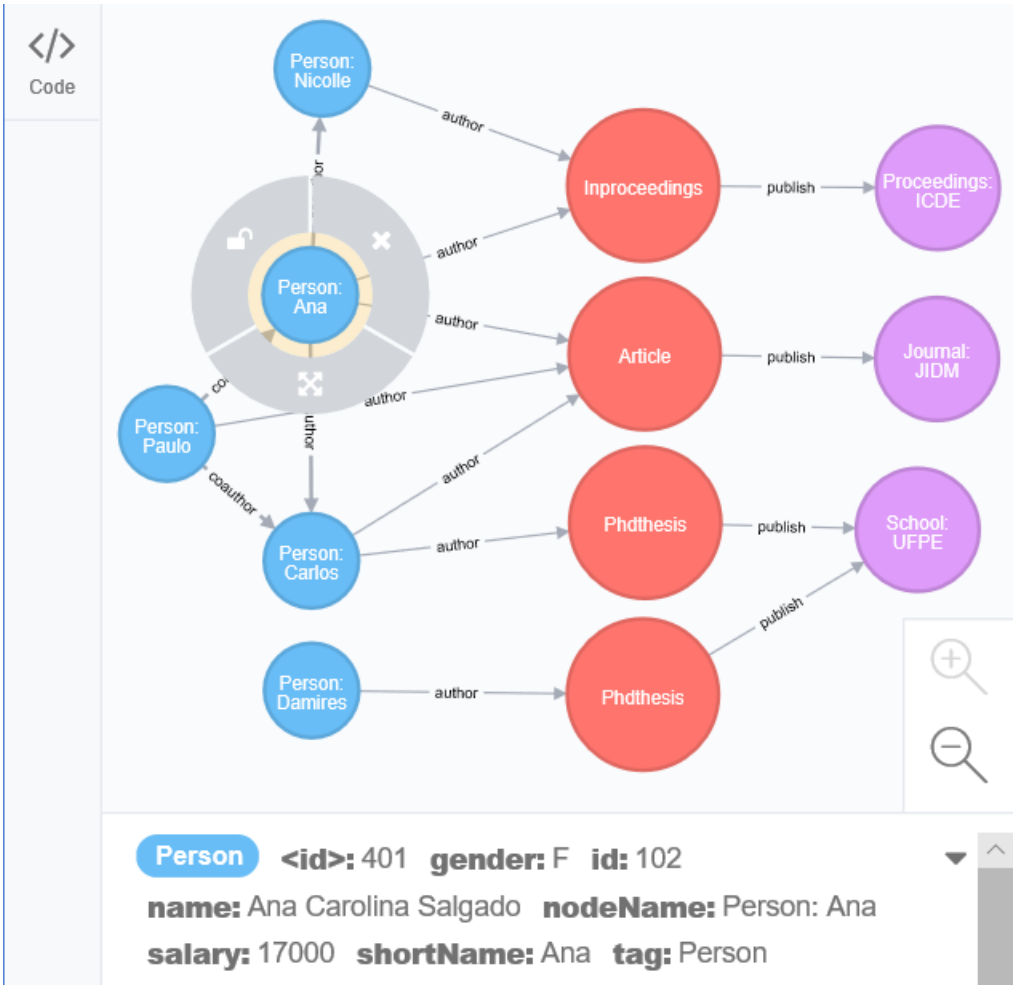


Figura 66 - Interface gráfica do Neo4j

Com base nesses recursos do Neo4j, reproduzimos os exemplos de consulta do Capítulo 5, utilizando as mesmas modelagens. Na explanação dos exemplos, apresentamos a representação gráfica da modelagem no Neo4j e, em seguida, mostramos as consultas parametrizadas e os seus respectivos resultados exibindo ilustrações da interface gráfica.

6.2.1 Consultas na modelagem MPGrafo-1

A Figura 67 apresenta, na interface gráfica do Neo4j, a modelagem MPGrafo-1. Nessa figura, o grafo atende à seguinte configuração visual: os VD são amarelos, mostrando o nome do rótulo e a propriedade principal, os VA são verdes, mostrando o nome do rótulo e os mesmos identificadores usados nos exemplos do Capítulo 5, e os SubGV, que correspondem aos vértices de rótulo (:Person), são azuis, mostrando o nome do rótulo e a propriedade principal.

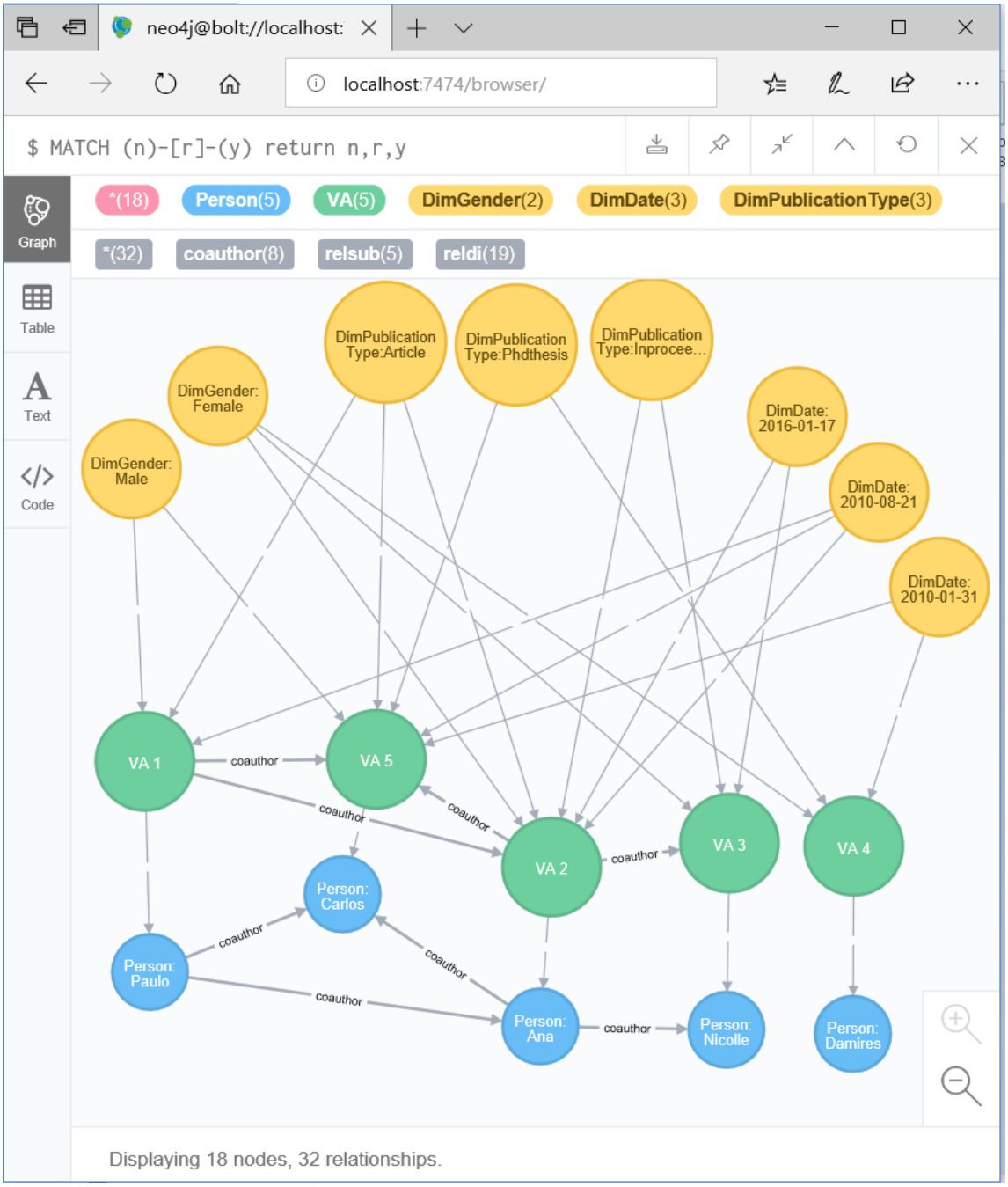


Figura 67 - Modelagem MPGrafo-1 no Neo4j

A partir dessa modelagem executamos as consultas no Neo4j e detalhamos as funcionalidades do sistema. Iniciamos com a reprodução do exemplo de consulta da Figura 54, que consulta e agrega os autores em função do ano de publicação. Essa consulta está expressa na Figura 68, mostrando: na linha 1 a chamada da função no Neo4j; na linha 2 a consulta em Cypher que seleciona a dimensão DimDate; na linha 3 as configurações de análise para os VA e os seus relacionamentos analíticos; na linha 4 as configurações de análise para os vértices de rótulo (:Person) que compõem o subgrafo agregado; e na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimDate:

'year'}.

1	Call olap.aggregate.graphResult(
2	"MATCH (dim:DimDate)-[reldi]->(va: r_{VA}), (: r_{VA})-[relan]->(: r_{VA}), (va: r_{VA})-[relsub: r_{relsub}]->(SubGV) RETURN reldi, relan, va, relsub;",
3	[{`*`:'count', betweenness:'collect', name:'collect', salary:'avg', degreeM:'min'}, {`*`:'count'}],
4	[{Person:{ salary:['sum','max'], degreeM:['sum','max']} }],
5	{DimDate:'year'})

Figura 68 - Consulta que agrega os autores pelos anos de publicação

Com a execução dessa consulta, obtivemos um resultado gráfico que retratamos nas Figuras 69 e 70. Esse resultado é representado por um grafo de propriedade, contendo: dois VD 2010 e 2016 da dimensão DimDate na cor amarela, três VAagr na cor verde, três SubGagr na cor cinza e cinco vértices (:Person) do grafo de dados na cor azul. Na agregação dos VAagr, o primeiro, da direita para esquerda, agregou três VA com publicações de 2010, o segundo agregou um VA com publicações de 2010 e 2016, e o terceiro agregou um VA com publicação de 2016. Nos resultados gráficos, cada VAagr possui o seu Subgrafo Agregado SubGagr, que no caso dessa modelagem corresponde à agregação dos vértice de rótulo (:Person).

O grafo de resposta também apresenta os resultados das medidas de análise, de modo que ao selecionar um vértice na interface gráfica aparece as suas propriedades com os valores resultantes das medidas. Assim, mostramos na Figura 69 duas imagens da mesma resposta, sendo que selecionamos na imagem A o VAagr com publicação de 2010 e na imagem B o VAagr com publicação de 2010 e 2016.

Na seleção desses vértices, a configuração de medida {`*`:'count', betweenness:'collect', name:'collect', salary:'avg', degreeM:'min'} descrita na linha 3 da Figura 68 produz as seguintes propriedades:

- count_* - informa a quantidade de VA agregado no VAagr;
- collect_ betweenness - lista os VA, que constituem o VAagr, e os seus respectivos valores no algoritmo *betweenness*;
- collect_name - lista os valores da propriedade “name” dos VA, que compõe o VAagr;
- avg_salary - apresenta a média dos valores da propriedade “salary”; e
- min_ degreeM: apresenta o menor valor da propriedade “degreeM”;

Os valores resultantes dessas medidas de análise são visíveis nas imagens da Figura 69.

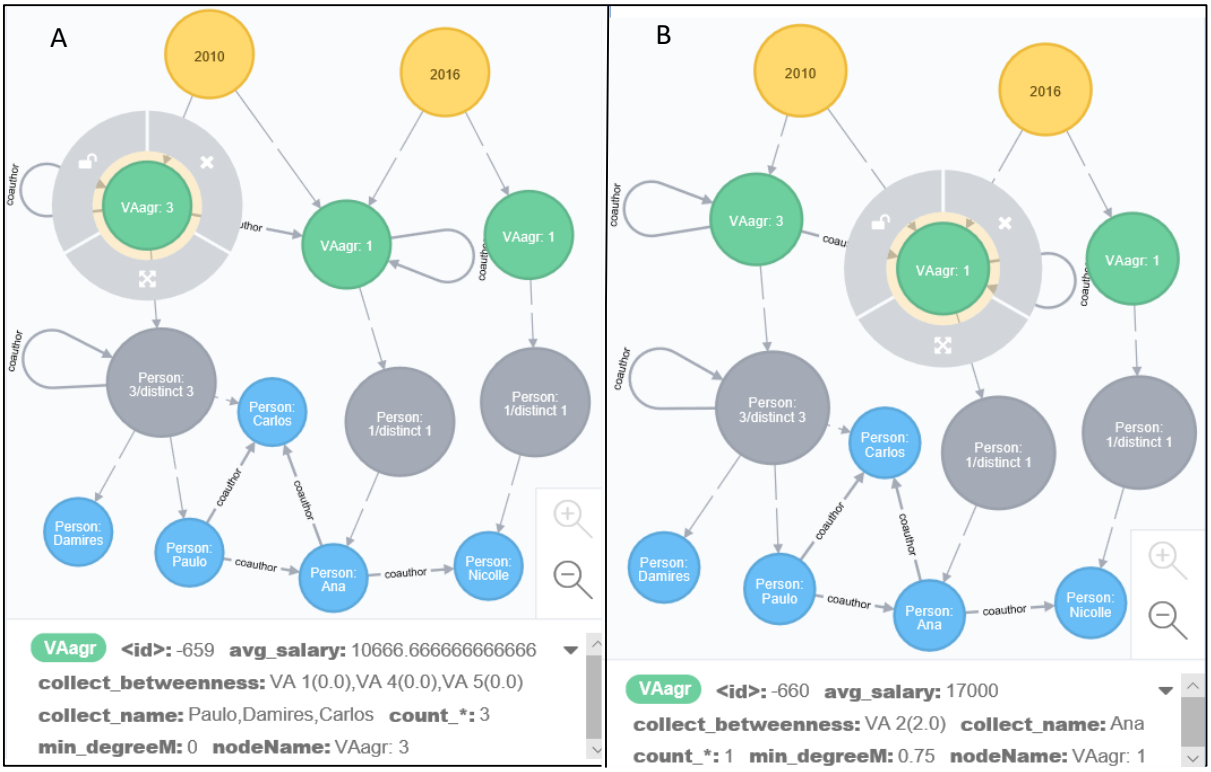


Figura 69 - Resultado da consulta que agrega os autores em função dos anos de publicação

Além desses resultados, a consulta também requisitou na linha 4 {Person:{ salary:['sum','max'], degreeM:['sum','max']} } da Figura 68 as medidas de análise para os vértices de rótulo (:Person) que constituem o subgrafo agregado SubGagr. Essa configuração de medida resultou nas seguintes propriedades:

- sum_salary - soma os valores da propriedade “salary”;
- max_salary - apresenta o maior valor da propriedade “salary”;
- sum_degreeM - soma os valores da propriedade “degreeM”;
- max_degreeM - apresenta o maior valor da propriedade “degreeM”;

Além dessas propriedades, definimos propriedades fixas para aparecer nos vértices do SubGagr, tais como:

- count_* - soma a quantidade de vértices na agregação;
- count_distinct - soma a quantidade de vértices diferentes na agregação.

Essas propriedades podem ser visualizadas ao selecionar um vértice do SubGagr, como mostra a Figura 70.

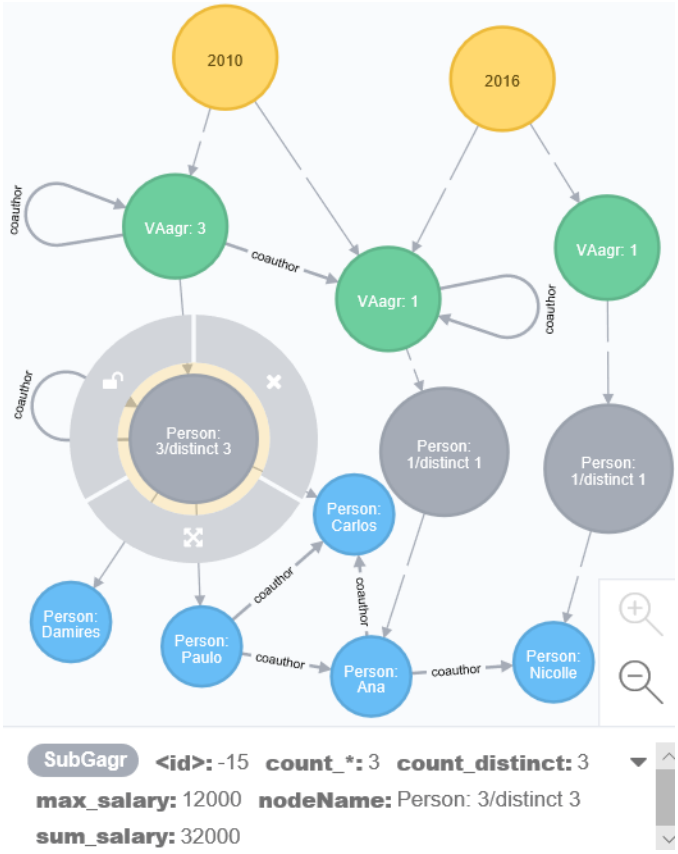


Figura 70 - Resultado da consulta selecionando um vértice do SubGagr

A próxima consulta é baseada no exemplo de consulta da Figura 56, que consulta e agrega os autores em função dos meses de publicação. Essa consulta equivale à operação de “Drill-down” na dimensão “DimDate” em relação à consulta da Figura 68, da mesma forma que a consulta da Figura 68 representa a operação contrária “Roll-up” em relação a essa. Com isso, dependendo das configurações da consulta é possível realizar determinadas operações OLAP.

1	Call olap.aggregate.graphResult(
2	"MATCH (dim:DimDate)-[reldi]->(va: r_{VA}), (: r_{VA})-[relan]->(: r_{VA}), (va: r_{VA})-[relsub: r_{relsub}]->(SubGV) RETURN reldi, relan, va, relsub;",
3	[{`*`:'count', closeness:['sum','collect'], name:'collect', degreeM:['sum','collect']}, {`*`:'count'}],
4	[],
5	{DimDate:'month'})

Figura 71 - Consulta que agrega os autores pelos meses de publicação

A Figura 71 apresenta a consulta especificando na linha 2 a mesma consulta em Cypher da Figura 68; na linha 3 as configurações de análise para os VA e os seus

relacionamentos analíticos; na linha 4 não definimos medidas de análise para os vértices do subgrafo agregado; e na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimDate: 'month'}.

Com a execução dessa consulta, obtivemos um resultado gráfico que retratamos na Figura 72. Esse resultado é representado por um grafo de propriedade contendo: três VD 2010-01, 2010-08 e 2016-01 da dimensão DimDate na cor amarela, cinco VAagr na cor verde, cinco SubGagr na cor cinza e cinco vértices (:Person) do grafo de dados na cor azul. Na agregação cada VAagr agregou apenas um VA, devido às diferentes combinações de VD que são relacionadas com os VA, como pode ser observado na Figura 72 cada VA se relacionando com distintas combinações de VD.

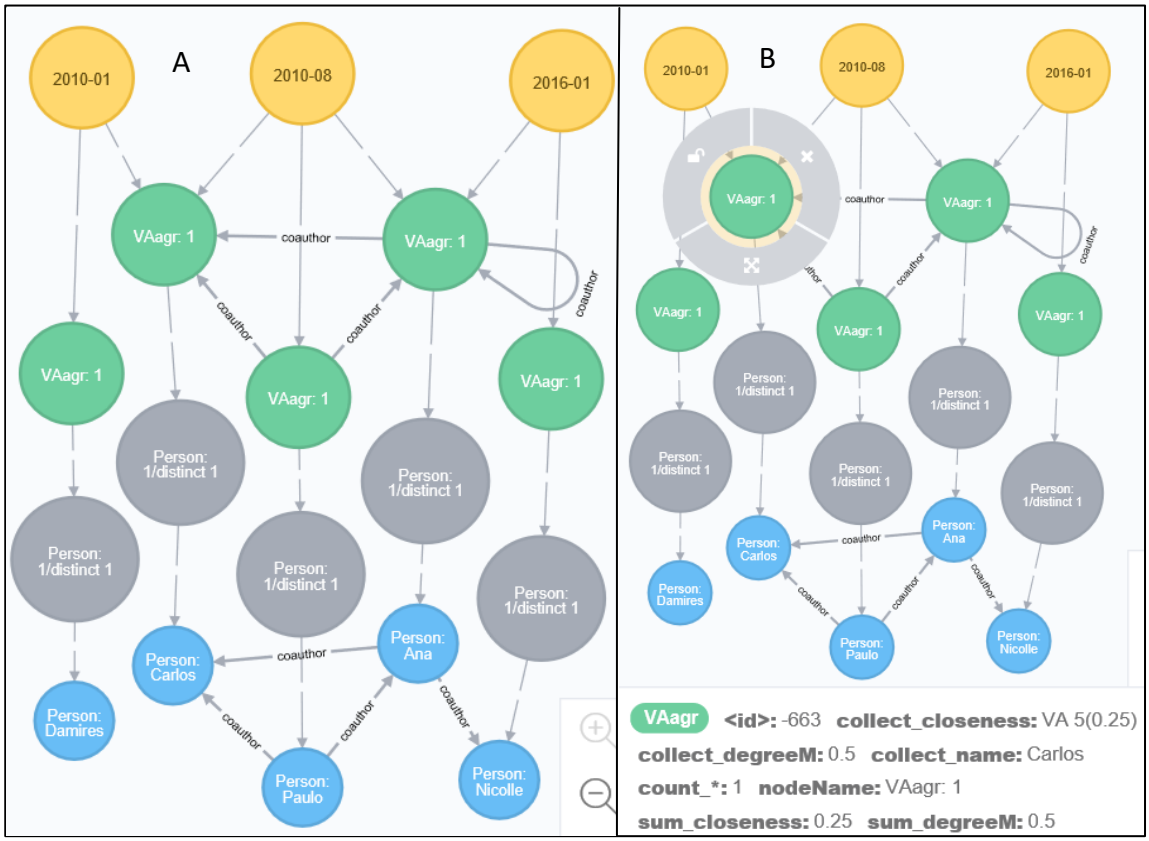


Figura 72 - Resultado da consulta que agrega os autores em função dos meses de publicação

Na exibição dos valores das medidas de análise, mostramos na imagem B da Figura 72 os resultados que correspondem ao VAagr que agregou um VA com publicações em 2010-01 e 2010-08. Para não ficar repetitivo, descrevemos apenas as propriedades da configuração de medida {`*`:'count', closeness:['sum','collect'], name:'collect', degreeM:['sum','collect']}, que ainda não foram contempladas nas consultas, tais como:

- sum_ closeness - soma os valores dos VA, que compõem o VAagr selecionado, no algoritmo closeness centrality;

- collect_ closeness - lista os VA, que compõem o VAagr selecionado, com os seus respectivos valores no algoritmo closeness centrality; e
- collect_ degreeM - lista os valores da propriedade “degreeM” dos VA, que compõe o VAagr selecionado;

Por fim, a imagem B na Figura 72 mostra os valores resultantes das medidas de análise nas respectivas propriedades.

A consulta seguinte é baseada no exemplo de consulta da Figura 57. Essa consulta corresponde à operação “Slice” na dimensão “DimDate”, recuperando apenas os autores com publicação em 2010, com a agregação dos autores em função do gênero.

1	Call olap.aggregate.graphResult(
2	"MATCH (dim_date:DimDate)-[reldi_date]->(va: r_{VA}), (dim_gender:DimGender)-[reldi_gender]->(va: r_{VA}), (: r_{VA})-[relan]->(: r_{VA}), (va: r_{VA})-[relsub: r_{relsub}]->(SubGV) where dim_date.year = '2010' RETURN reldi_date, reldi_gender, relan, va, relsub;",
3	[{`*`:'count', closeness:['collect'], betweenness:['collect'], name:'collect', salary:['sum','avg'] }, {`*`:'count'}],
4	[],
5	{DimDate:'year', DimGender:'gender'})

Figura 73 - Consulta os autores com publicação em 2010 e pelos meses de publicação

A Figura 73 apresenta a consulta especificando na linha 2 uma consulta em Cypher que seleciona as dimensões DimDate e DimGender, e restringe a DimDate com a expressão “dim_date.year = '2010'”; na linha 3 as configurações que analisam os VA e os seus relacionamentos analíticos; na linha 4 não definimos medidas de análise para os vértices do subgrafo agregado; e na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimDate:'year', DimGender:'gender'} .

Com a execução dessa consulta, obtivemos um resultado gráfico que retratamos na Figuras 74. Esse resultado é representado por um grafo de propriedade contendo: três VD na cor amarela, sendo um da dimensão DimDate 2010 e dois da dimensão DimGender “M” e “F”, dois VAagr na cor verde, dois SubGagr na cor cinza e quatro vértices (:Person) do grafo de dados na cor azul. Na agregação cada VAagr agregou dois VA considerando os relacionamentos entre os VA e os VD. Nessa consulta os 4 autores se relacionam com o VD 2010, sendo que dois deles se relacionam também com os VD “M” e os outros 2 com os VD “F”. Com esses relacionamentos, são formadas duas combinações que produzem

os dois VAagr resultantes, como pode ser observado na Figura 74.

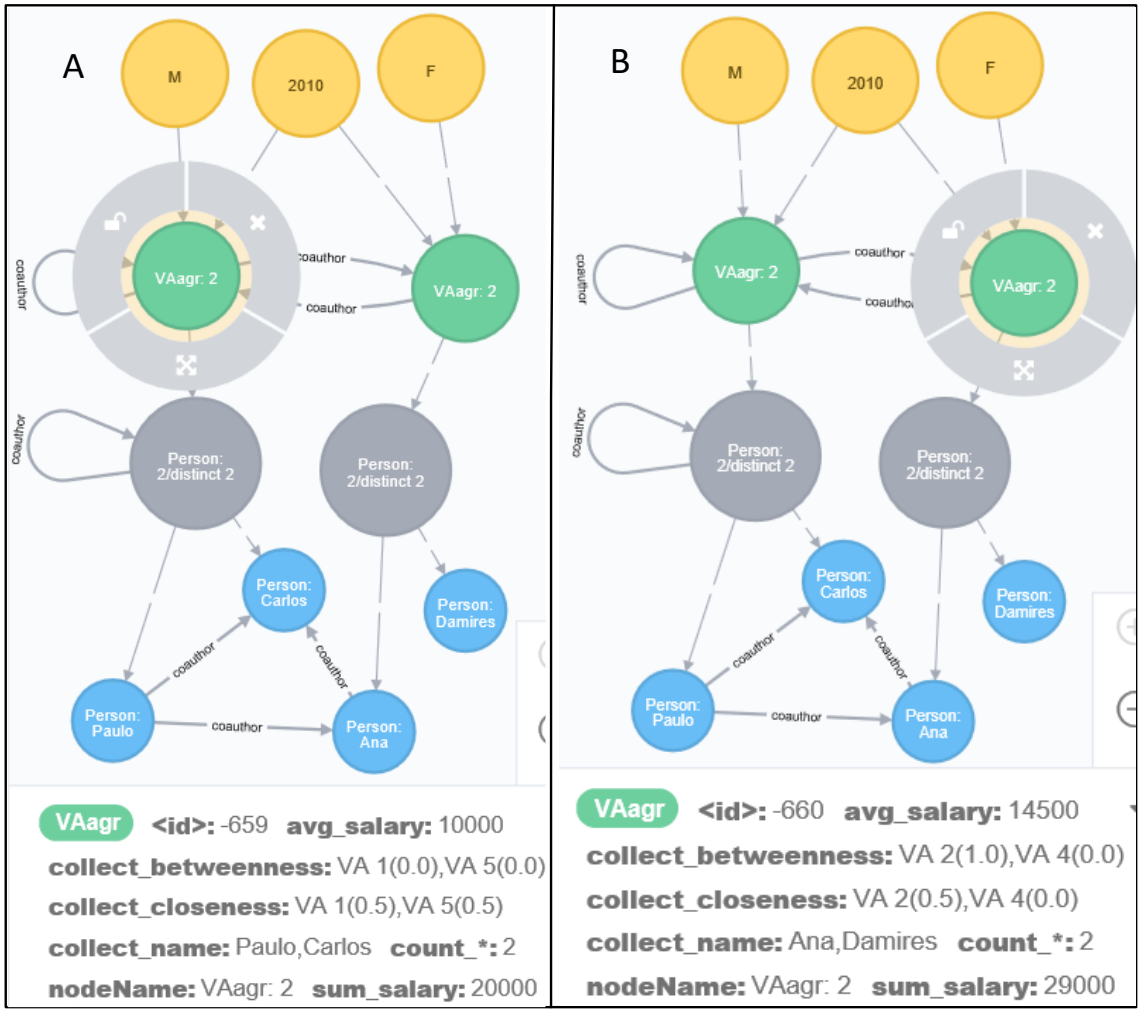


Figura 74 - Resultado da consulta que agrega pelo gênero os autores com publicação em 2010

Para exibir os valores das medidas de análise, mostramos na Figura 74 duas imagens sobre o mesmo grafo de resposta, sendo que na imagem A, selecionamos o VAagr composto pelos VA do gênero masculino com publicação em 2010 e na imagem B, selecionamos o VAagr composto pelos VA do gênero feminino com publicação em 2010. Nessas imagens aparecem os resultados da configuração de medida descrita na linha 3 da Figura 73. Os valores resultantes das medidas de análise são visíveis nas propriedades das imagens A e B da Figura 74. Assim, finalizamos as consultas sobre a modelagem MPGrafo-1.

6.2.2 Consultas na modelagem MPGrafo-2

A Figura 75 apresenta a modelagem MPGrafo-2 na interface gráfica do Neo4j. Nessa figura o grafo atende à seguinte configuração visual: os VD são amarelos, mostrando o nome do rótulo e a propriedade principal, os VA são verdes, mostrando o nome do rótulo e

o identificador, e os SubGV mostram o nome do rótulo e a propriedade principal em todos os seus vértices, os quais possuem três cores: azul para os vértices de rótulo (:Person), vermelho para os vértice de rótulo (:Publication) e roxo para os vértices de rótulo (:Venue), como ilustra a Figura 75.

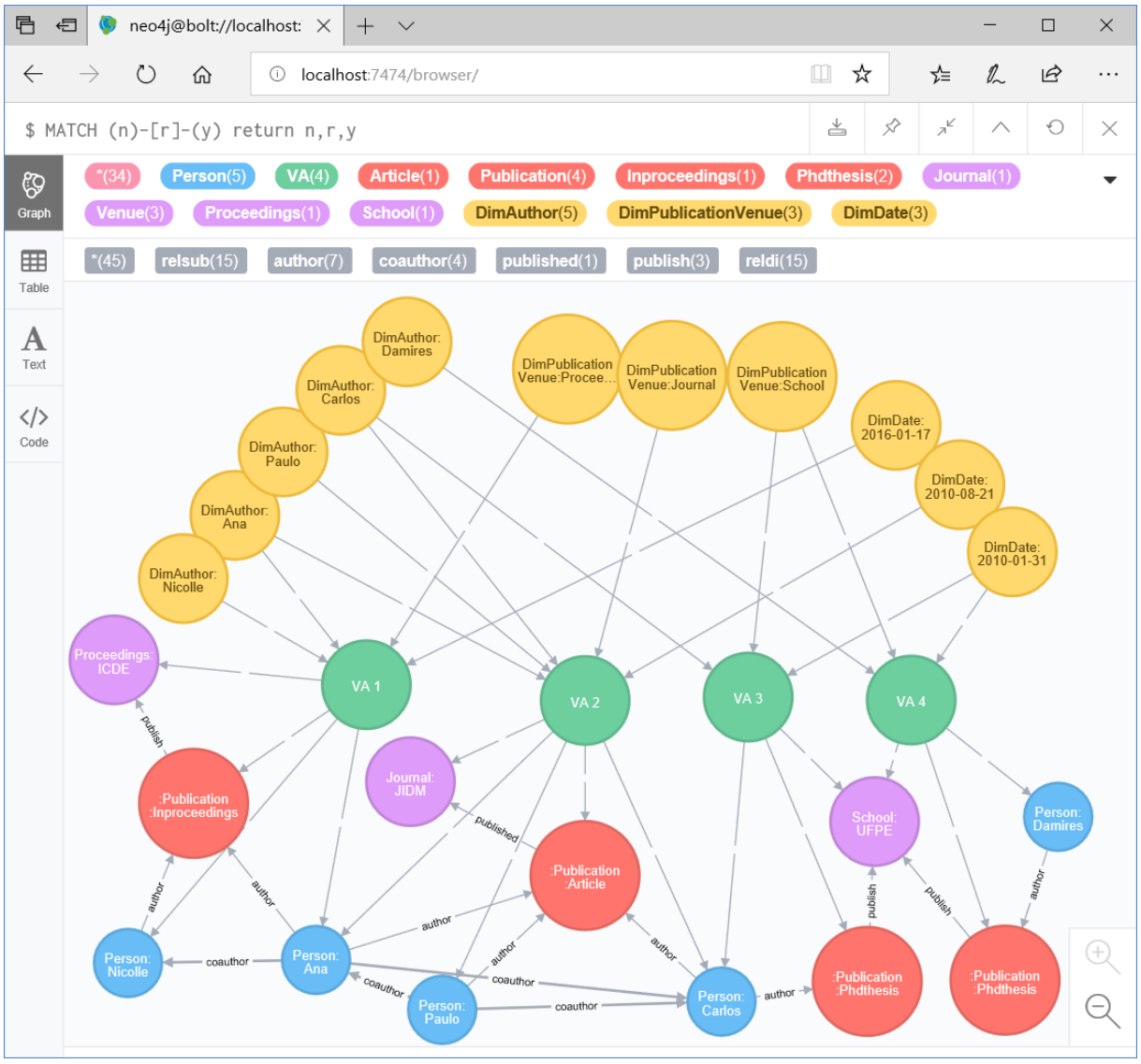


Figura 75 - Modelagem MPGrafo-2 no Neo4j

A partir dessa modelagem executamos as consultas no Neo4j, iniciando com o exemplo das Figuras 59 e 60, que agrega as informações de publicações em função do ano e local de publicação. A Figura 76 apresenta a consulta especificando na linha 2 uma consulta em Cypher que seleciona as dimensões DimDate e DimPublicationVenue; na linha 3 as configurações que analisam os VA; na linha 4 não foram requisitadas medidas de análise para analisar os vértices do subgrafo agregado; e na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimDate:'year',DimPublicationVenue:'type'}.

```
1  Call olap.aggregate.graphResult(
2  "MATCH (dim_date:DimDate)-[reldi_date]->(va: r_VA), (va: r_VA)-[relsub: r_relsub]->(SubGV), (dim_publ:DimPublicationVenue)-[reldi_publ]->(va: r_VA) RETURN reldi_date, reldi_publ, va, relsub;",
3  [{`*`:'count', citation:['sum','avg'], author:['sum'], sumSalary:['sum','avg'], sumDegreeM:['sum','avg']}, {`*`:'count'}],
4  [],
5  {DimDate:'year',DimPublicationVenue:'type'})
```

Figura 76 - Consulta as informações de publicação agregando-as por ano e local de publicação

Com a execução dessa consulta, obtivemos um resultado gráfico que retratamos na Figura 77. Esse resultado é representado por um grafo de propriedade, contendo: cinco VD na cor amarela, sendo dois da dimensão DimDate 2010 e 2016, e três da dimensão DimPublicationVenue “School”, “Journal” e “Proceedings”; três VAagr na cor verde; nove SubGagr na cor cinza, visto que em cada subgrafo analítico existem três rótulos diferentes; cinco vértices de rótulo (:Person) na cor azul; quatro vértices de rótulo (:Publication) na cor vermelha; e três vértices de rótulo (:Venue) na cor roxa.

Na agregação dos VA foram produzidos três VAagr: o primeiro, da esquerda para direita, agregou dois VA e possui a combinação dos VD “School” e 2010; o segundo agregou um VA e possui a combinação dos VD “Journal” e 2010; e o terceiro agregou um VA e possui a combinação dos VD “Proceedings” e 2016.

O grafo de resposta representado na Figura 77 mostra selecionado o VAagr com a combinação de VD “Journal” e 2010. Com isso, aparecem os resultados da configuração de medida descrita na linha 3 da Figura 76 {`*`:'count', citation:['sum','avg'], author:['sum'], sumSalary:['sum','avg'], sumDegreeM:['sum','avg'], que resultaram nas seguintes propriedades:

- count_* - informa a quantidade de VA, que compõem o VAagr selecionado;
- sum_citation - soma os valores da propriedade “citation”;
- avg_citation - calcula a média dos valores da propriedade “citation”;
- sum_author - soma os valores da propriedade “author”;
- sum_sumSalary - soma os valores da propriedade “sumSalary”;
- avg_sumSalary - calcula a média dos valores da propriedade “sumSalary”;
- sum_sumDegreeM - soma os valores da propriedade “sumDegreeM”; e

- avg_sumDegreeM - calcula a média dos valores da propriedade “sumDegreeM”;

Os valores resultantes das medidas são visíveis na Figura 77.

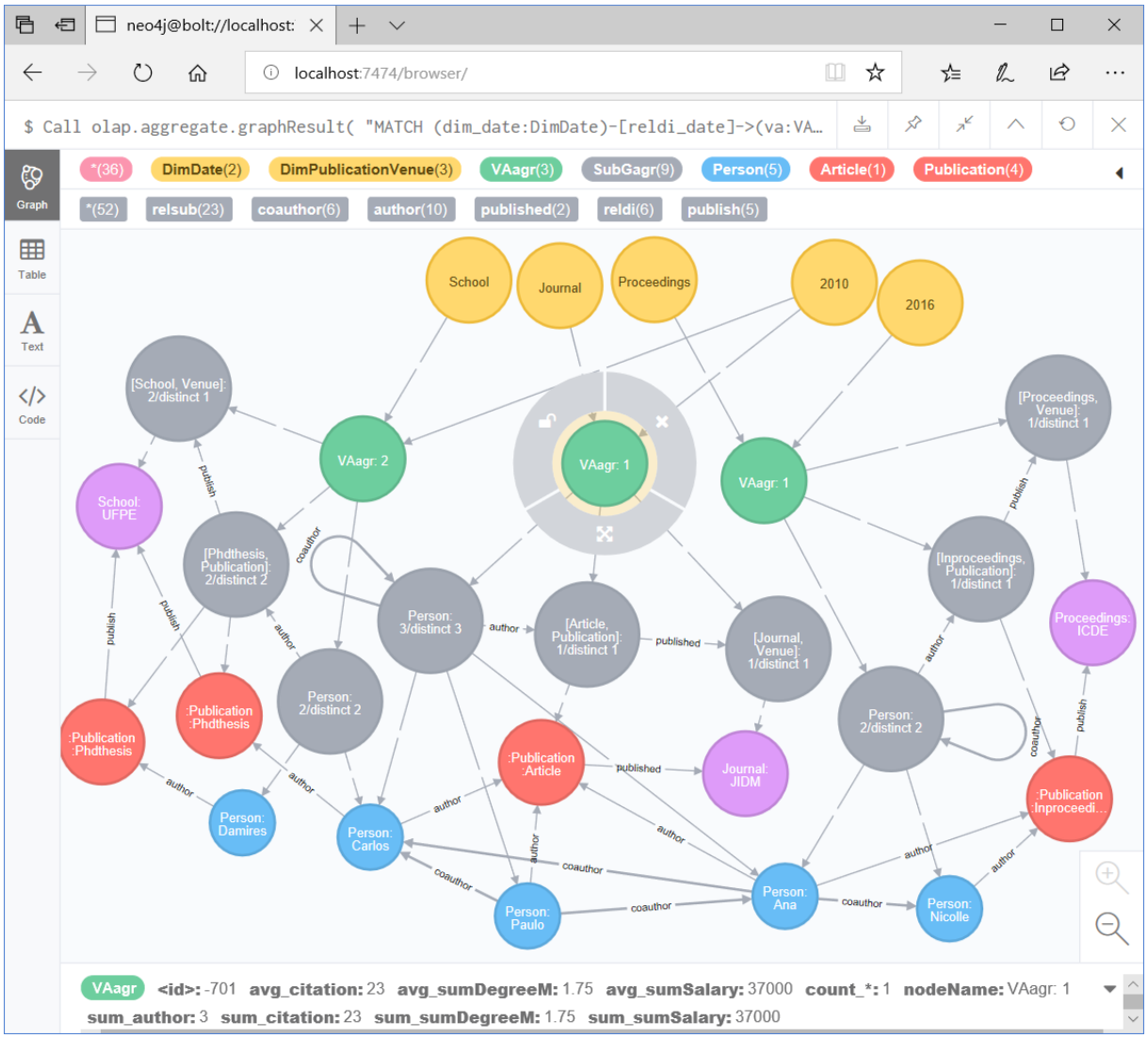


Figura 77 - Resultado da consulta que agrega as publicações por ano e local de publicação

A próxima consulta é baseada no exemplo de consulta das Figuras 61 e 62, que agrega as informações de publicações que foram realizadas nas instituições (School) no ano 2010. A Figura 78 apresenta a consulta especificando na linha 2 uma consulta em Cypher que seleciona e restringe as dimensões com os respectivos valores: DimDate com a propriedade “year” = 2010 e DimPublicationVenue com a “type” = “School”; na linha 3 as configurações que analisam os VA; na linha 4 definimos medida de análise para os vértices de rótulo {:Person, :Publication e :Venue} que compõem o Subgrafo Analítico; e na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimDate:'year',DimPublicationVenue:'type'} .

Essa consulta corresponde à aplicação da operação “Dice” sobre a consulta da Figura

76. Ela restringe os VD da dimensão “DimDate” recuperando apenas os VD com valor “2010” na propriedade “year”, e os VD da dimensão “DimPublicationVenue” com valor “School” na propriedade “type”. Com a execução dessa consulta, obtivemos um resultado que é ilustrado na Figura 79.

```
1  Call olap.aggregate.graphResult(
2  "MATCH (dim_date:DimDate)-[reldi_date]->(va: r_VA),(dim_publ:DimPublicationVenue)-[reldi_publ]->(va: r_VA),(va: r_VA)-[relsub: r_relsub]->(SubGV) where dim_date.year = '2010' and dim_publ.type = 'School' RETURN reldi_date, reldi_publ, va, relsub;",
3  [{`*`:'count', citation:['sum','avg'], author:['sum']}, {`*`:'count'}]
4   [{Person:{salary:['collect','max'],shortName:['collect']},
    Publication:{citation:['collect']}, Venue:{shortName:['collect']} }],
5  {DimDate:'year',DimPublicationVenue:'type'})
```

Figura 78 - Consulta e agrega as informações de publicações das instituições no ano 2010

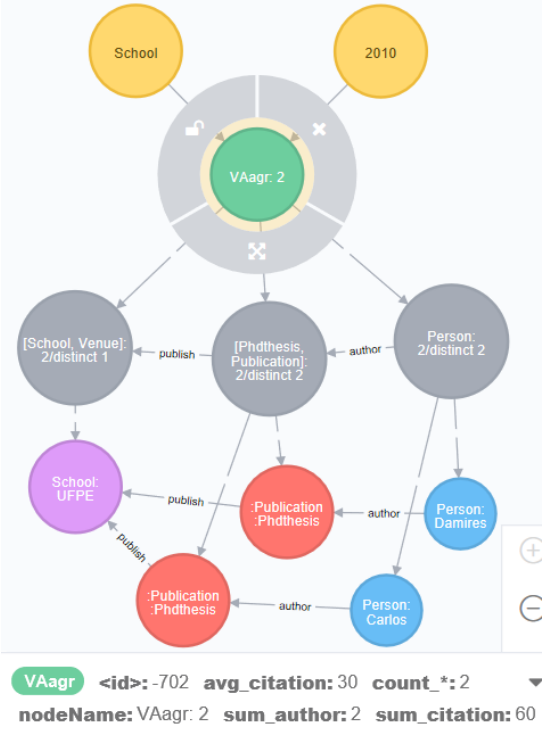


Figura 79 - Resultado da consulta que agrega as publicações de instituições em 2010

Esse resultado da consulta apresenta um grafo de propriedades, contendo: dois VD na cor amarela, sendo um da dimensão DimDate com valor 2010 e outro da dimensão DimPublicationVenue com valor “School”; um VAagr na cor verde; três SubGagr na cor cinza; dois vértices de rótulo (:Person) na cor azul; dois vértices de rótulo (:Publication) na

cor vermelha; um vértices de rótulo (:Venue) na cor roxa.

Na agregação dos VA, foi produzido um único VAagr, que agregou dois VA combinando os VD “School” e 2010, como mostra a Figura 79. Além disso, essa figura apresenta o VAagr selecionado mostrando nas propriedades os valores resultantes das medidas de análise, que foram requisitadas na linha 3 da Figura 78.

Além desses resultados, essa consulta também requisitou na linha 4 da Figura 78 medidas de análise para processar os vértices de rótulo Person, Publication e Venue que constituem o SubGagr. A configuração de medida [{Person:{salary:['collect','max'], shortName:['collect']}, Publication:{citation:['collect']}, Venue:{shortName:['collect']} }] resultou em valores e propriedades que retratam as medidas de análise nos vértices do SubGagr, como mostram os valores dos vértices selecionados na Figura 78.

Essas medidas são expressas nas propriedades dos seguintes rótulos:

- Person
    - collect_salary - lista os valores da propriedade “salary” dos vértices (:Person);
    - max_salary - apresenta o maior valor da propriedade “salary” dos vértices (:Person);
    - collect_ shortName - lista os valores da propriedade “shortName” dos vértices (:Person);
- Publication
    - collect_ citation - lista os valores da propriedade “citation” dos vértices (:Person);
- Venue
    - collect_ shortName - lista os valores da propriedade “shortName” dos vértices (:Person);

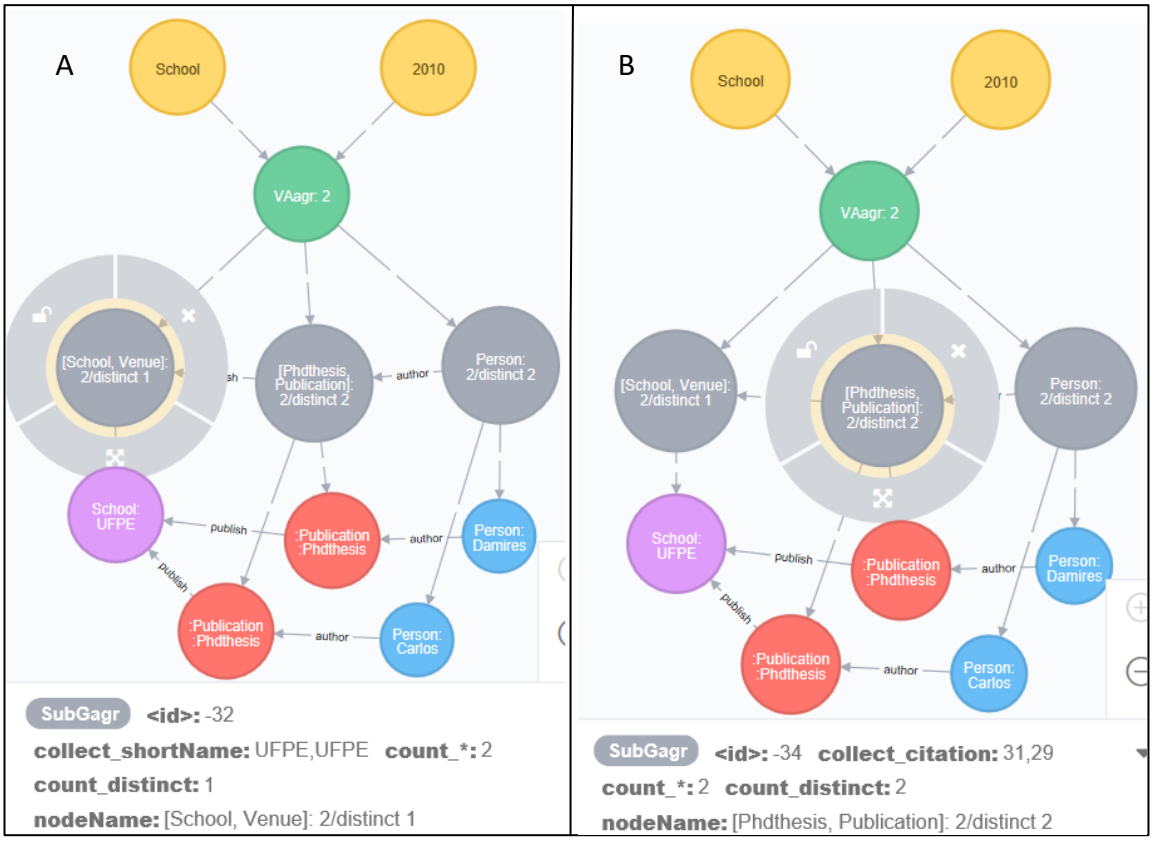


Figura 80 - Resultado da consulta detalhando os vértices do SubGagr

Por fim, concluímos a apresentação das consultas multidimensionais em grafo utilizando a implementação da AAMPGrafo no Neo4j. Na próxima seção, discutimos a representação visual das consultas em Cypher e a forma de representar os dados agregados em consultas analíticas e multidimensionais em grafo.

6.3 VISUALIZAÇÃO DE CONSULTAS AGREGADAS

Tradicionalmente, as consultas analíticas são representadas em tabelas com valores agregados. Neste trabalho, não consideramos essa forma de representação a mais indicada para lidar com grafo de dados, visto que o valor do conteúdo e a topologia do grafo são ambos importantes para a análise. Na revisão sistemática, percebemos que a definição do grafo agregado tem sido utilizada para produzir cuboids e representar agregação de dados em grafo. No entanto, não encontramos trabalhos que utilizassem grafo agregado e medidas de análise (valores agregados) juntas em uma interface gráfica. Com base nesse contexto, introduzimos uma representação em interface gráfica dos resultados de consultas multidimensionais em grafo. Essa representação apresenta um grafo agregado com resultados de medidas de análise em uma estrutura orientada aos assuntos das dimensões.

No Neo4j e nos artigos relacionados que possuem uma visualização de consulta

(JAKAWAT et al., 2016b), o grafo agregado e os valores agregados das medidas são apresentados separadamente. Por exemplo, o trabalho de JAKAWAT; FAVRE; LOUDCHER (2016a) apresenta a estrutura topológica do grafo separada dos valores agregados dos dados, como observado na Figura 81. Além disso, na pesquisa mencionada, a interface foi desenvolvida de forma específica para os dados utilizados, visto que as consultas são definidas na implementação sem a possibilidade de alterá-las ou de executar novas consultas.

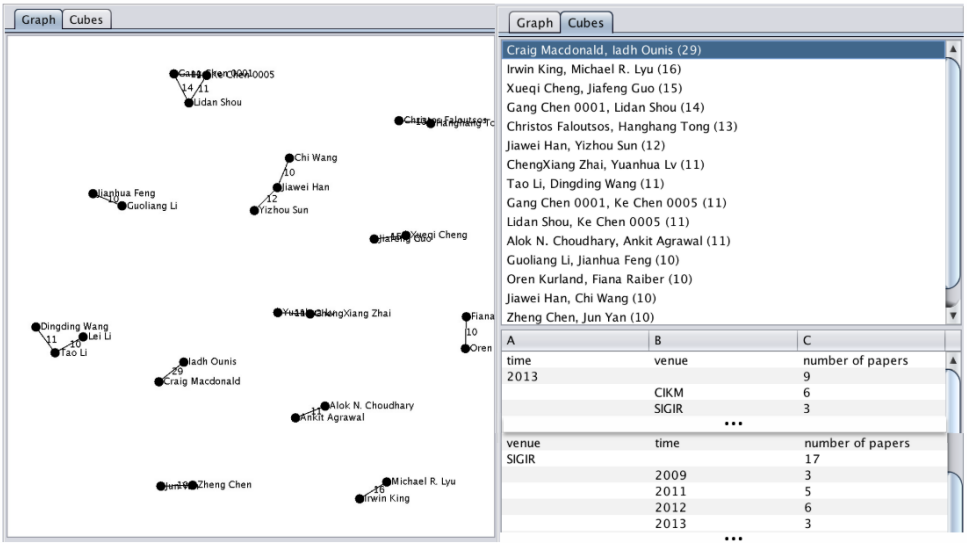


Figura 81 - Interface gráfica do trabalho de JAKAWAT; FAVRE; LOUDCHER (2016a)

No Neo4j a linguagem Cypher contém funcionalidades que permitem aplicar funções de agregação (sum, avg, count) sobre os valores das propriedades do grafo, desconsiderando a estrutura topológica do grafo. A interface gráfica do Neo4j permite apresentar em consultas separadas o valor agregado das propriedades e a topologia dos dados em grafo, como mostra a Figura 82.

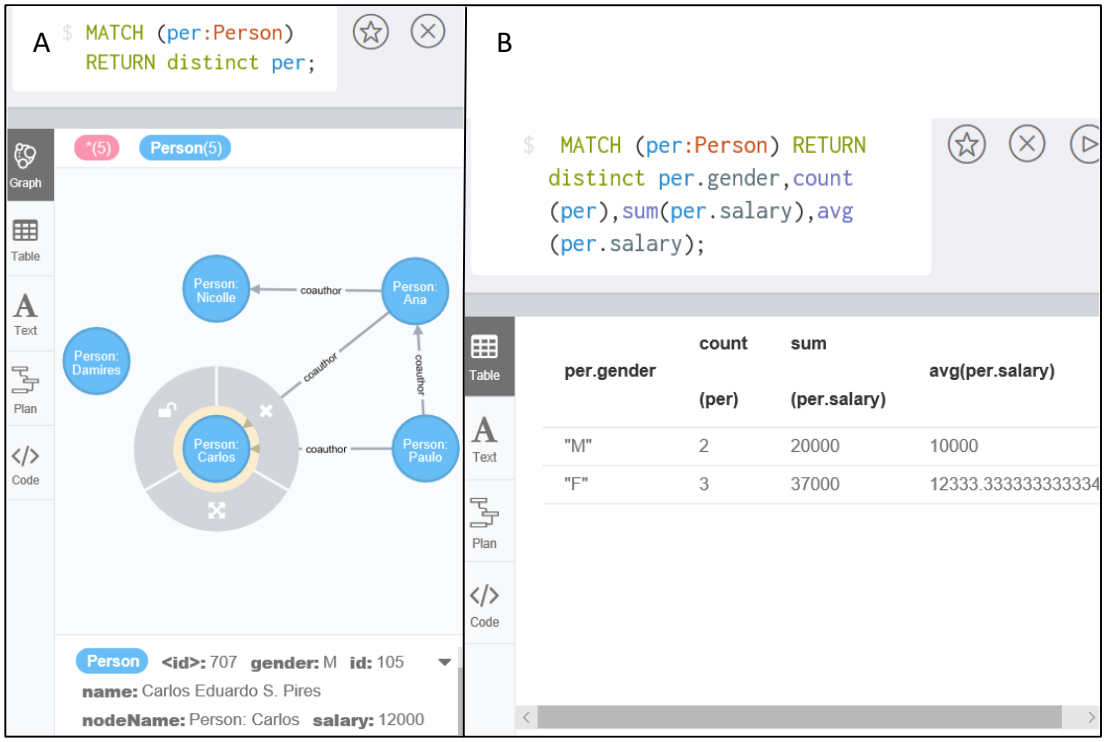


Figura 82 - Visualização da consulta em Cypher

Na Figura 82A, a consulta foi executada em 1ms e resultou em um grafo com os vértices (:Person) da amostra DBLP. Já na Figura 82B, a consulta foi executada em 2ms e resultou em uma tabela com dados agregados. Essa tabela agrupou os vértices (:Person) em função do gênero, computando a quantidade, a soma e a média dos salários pelos gêneros das pessoas. Com isso, consideramos que a visualização do resultado da análise é fragmentada, separando valores agregados da topologia do grafo. Além disso, a visualização desses resultados não apresenta a agregação da topologia do grafo de dados.

Na AAMPGrafo, as respostas das consultas expressam de forma unificada tanto a agregação de padrões em grafo quanto os resultados de medidas de análise. Essa representação visual poder ser considerada nova, visto que não encontramos na revisão sistemática indícios de uma representação multidimensional de grafo agregado que associe a agregação topológica do grafo de dados com os assuntos (perspectivas) das dimensões. A Figura 73 mostra a representação visual para a resposta de consultas multidimensionais em grafo.

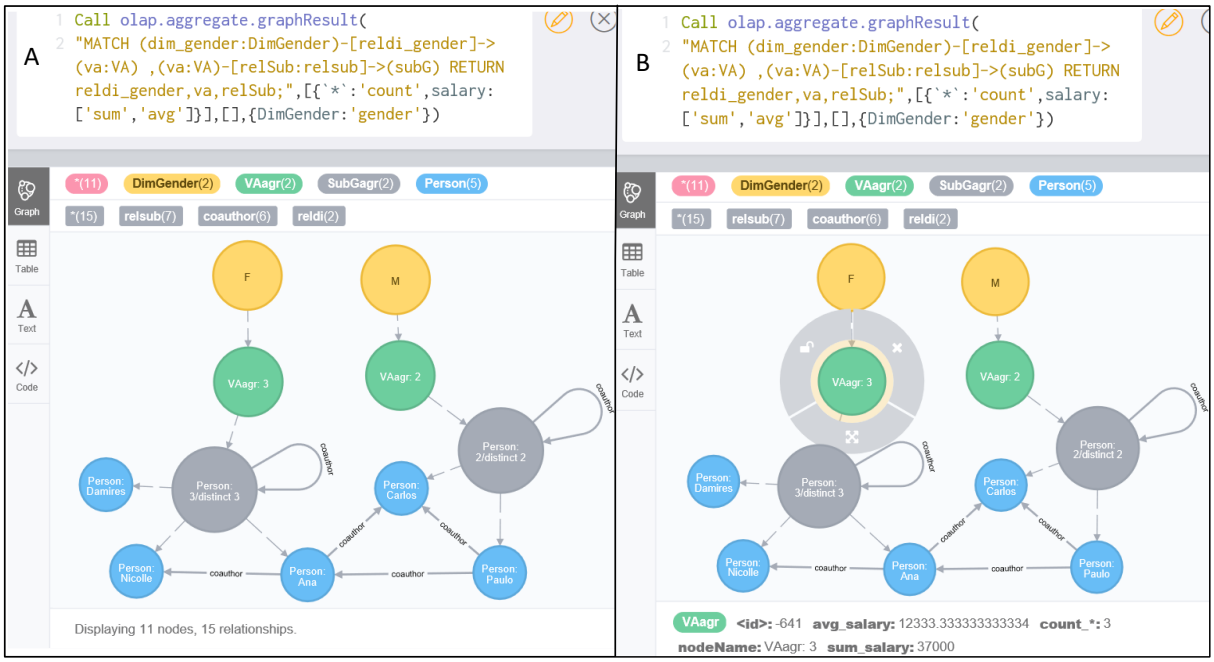


Figura 83 - Visualização da consulta na AAMPGrafo

A consulta da Figura 83 foi realizada em 4ms e apresenta um resultado correspondente à consulta da Figura 82B acrescido com a representação gráfica do grafo agregado que propomos. Essa forma de representar é uma contribuição no campo do Grafo BI e áreas afins com análises multidimensionais em grafo, pois define uma estrutura orientada a assunto que possibilita visualizar as dimensões sobre a topologia do grafo contendo os valores agregados das medidas de análise.

Diante do exposto, os trabalhos observados para a visualização de consultas multidimensionais em grafo apresentam a estrutura topológica separada dos valores agregados. O nosso trabalho, por sua vez, especifica os grafos agregados aos assuntos das dimensões e, além disso, esses grafos agregados contêm seus respectivos valores agregados das medidas de análise. Na próxima seção, apresentamos um experimento da AAMPGrafo utilizando uma amostragem com grande volume de dados.

6.4 EXPERIMENTOS DA AAMPGRAFO

No experimento deste trabalho, observamos o comportamento da AAMPGrafo na execução de consultas multidimensionais em grandes grafos de dados. Para compor esses grafos de dados, extraímos do repositório DBLP os dados referentes a 31885 publicações e produzimos um base de dados em grafo. Essa base de dados atende ao esquema da Figura 23, o qual foi utilizado na exemplificação das consultas, e possui os seguintes vértices de dados:

- 27807 vértices de autor com o rótulo {:Person};
- 9835 vértices de publicação com os rótulos {:Publication e :Article};
- 21960 vértices de publicação com os rótulos {:Publication e Inproceedings};
- 90 vértices de publicação com os rótulos {:Publication e Phdthesis};
- 992 vértices do local de publicação com os rótulos {:Venue e :Journal};
- 58 vértices do local de publicação com os rótulos {:Venue e :School}; e
- 3393 vértices do local de publicação com os rótulos {:Venue e :Proceedings};

Com esses dados, produzimos um base de dados em grafo com 64135 vértices e 208996 arestas. A Figura 84 mostra o esquema que representa os grafos de dados formados, selecionando o vértice de rótulo :Person para indicar a quantidade de autores no grafo count = 27807.

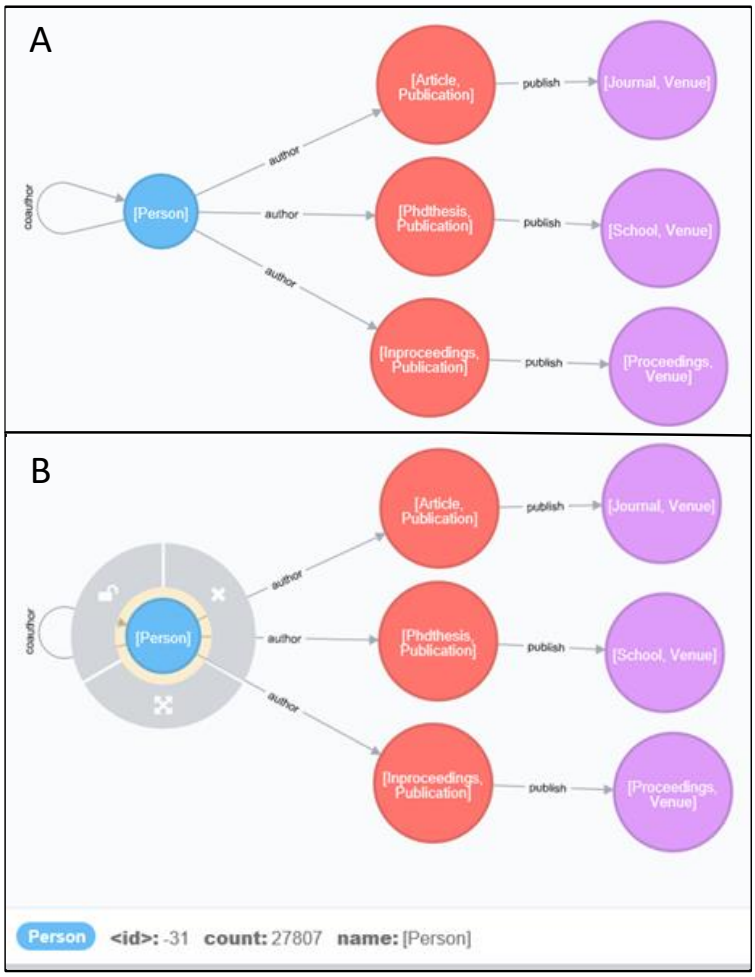


Figura 84 - Esquema do grafo de dados com 31885 publicação

Utilizando a base de dados gerada e as especificações do modelo MPGrafo, produzimos a modelagem MPGrafo-3 composta por duas dimensões {DimDate e DimPublicationType}. A Figura 85 mostra o esquema dessa modelagem, destacando a

quantidade de vértices que foram inseridos à base de dados para tornar os grafos de dados adequados ao modelo MPGrafo. Com isso, detalhamos a seguir os vértices inseridos na modelagem:

- 27807 Vértices Analíticos VA para representar os autores do grafo de dados;
- 3 Vértices Dimensionais da dimensão DimPublicationType para representar os tipos de publicação {:Article ou :Phdthesis ou :Inproceedings}; e
- 4412 Vértices Dimensionais da dimensão DimDate para representar as data das publicações.

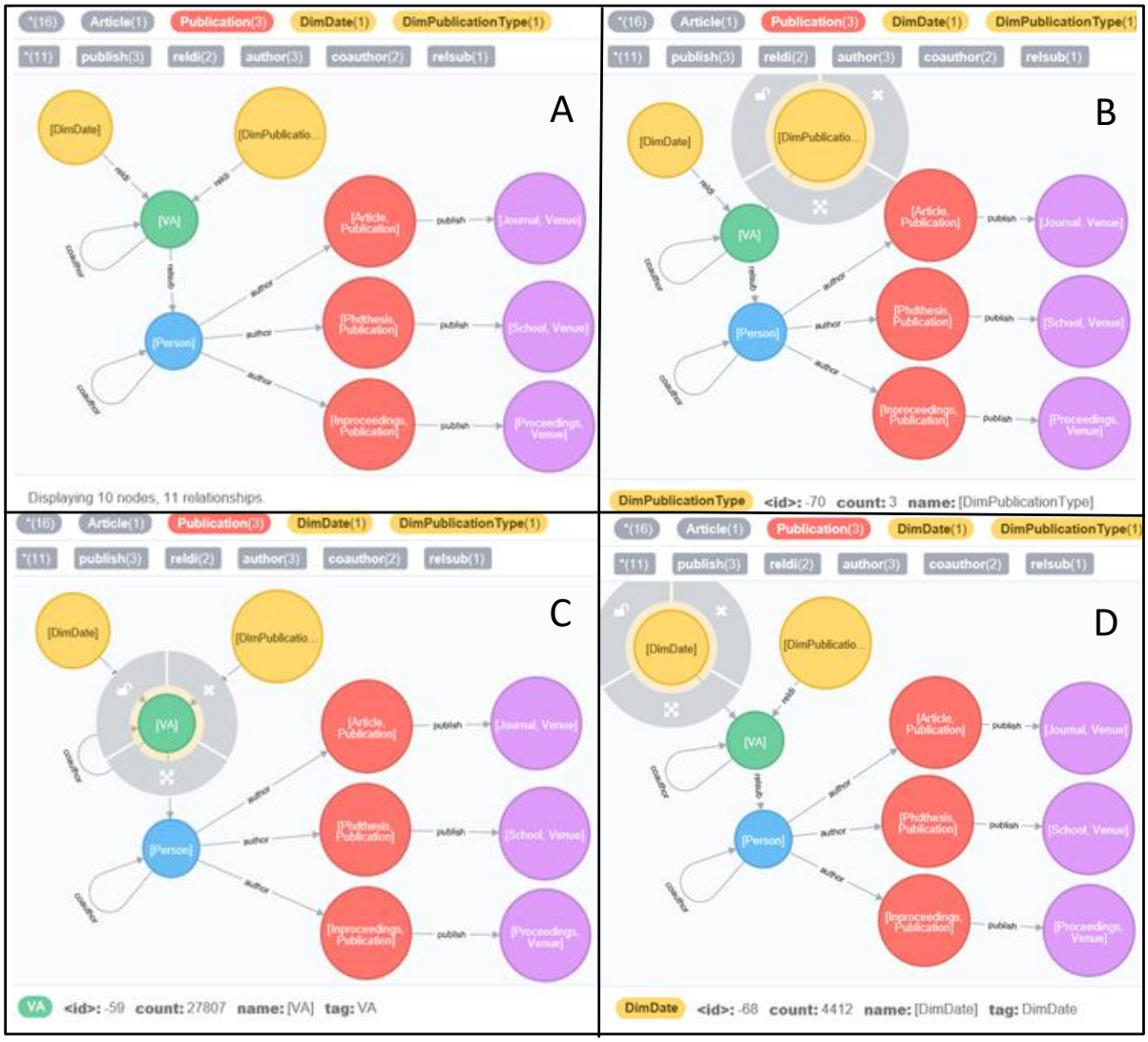


Figura 85 - Esquema da modelagem MPGrafo-3 com 31885 publicação

Na experimentação da AAMPGrafo, realizamos os experimentos de consulta em uma máquina com o sistema operacional Windows 10, o processador Intel(R) Core(TM) i7-7500 CPU 2.9 GHz e a memória RAM de 16 GB. As execuções das consultas foram realizadas no SGBDG Neo4j, da versão neo4j-community-3.3.4, com a instalação do *plugin* que

desenvolvemos.

Durante o experimento, utilizamos a modelagem MPGrafo-3 para executar diferentes consultas e observar o comportamento da AAMPGrafo sobre um grande grafo de dados. As consultas realizadas na experimentação testam as formas de análise, combinando dimensões, medidas analíticas e medidas topológicas na elaboração das consultas.

```
1   Call olap.aggregate.graphResult(
2   "MATCH (dim_publ:DimPublicationType)-[reldi_publ]->(va:VA),(va:VA)-[relSub:relsub]->(subG)  RETURN reldi_publ, va, relSub;",
3   [{`*`:'count',salary:['avg'],degreeM:['avg']}],
4   [],
5   {DimPublicationType:'type'},
6   ,{graphAggregate:false})YIELD dimensionalNodes,AggregEventNodes,relationships
```

Figura 86 - Consulta que agrega os autores pelos tipos de publicações realizadas

Na inicialização do experimento, elaboramos um consulta que utiliza apenas a dimensão DimPublicationType para analisar os autores em função dos tipos de publicações, agregando, assim ,os autores que possuem os mesmos tipos de publicação. A Figura 86 apresenta essa consulta especificando na linha 2 a consulta em Cypher que seleciona a dimensão DimPublicationType; na linha 3 as configurações de análise para os VA; na linha 4 não definimos medidas de análise para os vértices do subgrafo agregado; na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimPublicationType:'type'}; e na linha 6 definimos em Cypher uma adaptação para atender um limitação visual contida no Neo4j, no qual permiti visualizar o grafo de respostas com até 300 vértices.

Com a execução dessa consulta, obtivemos o resultado gráfico em 2769 ms e o retratamos na Figura 87. Esse resultado é composto por um grafo de propriedade contendo: três VD {Inproceedings, Phdthesis e Article} da dimensão DimPublicationType na cor amarela e seis VAagr na cor verde. Observando os dados agregados, identificamos que 13699 autores possuem apenas publicações do tipo Inproceedings; só 1 autor possui apenas publicações dos tipos {Phdthesis e Article}; e 84 autores possuem publicações dos três tipos {Inproceedings, Phdthesis e Article}.

O grafo expresso na resposta também apresenta os resultados das medidas de análise, de modo que ao selecionar um vértice na interface gráfica aparece as suas

propriedades com os valores resultantes das medidas. Assim, mostramos na Figura 87 quatro imagens da mesma resposta para apresentar os valores das medidas que foram especificadas na linha 3 da Figura 86 e são representadas nas seguintes propriedades:

- count_* - informa a quantidade de VA agregado no VAagr;
- avg_salary - apresenta a média dos valores da propriedade “salary”; e
- avg_ degreeM: apresenta a média dos valores da propriedade “degreeM”;

Os valores resultantes dessas medidas de análise são visíveis na Figura 87.

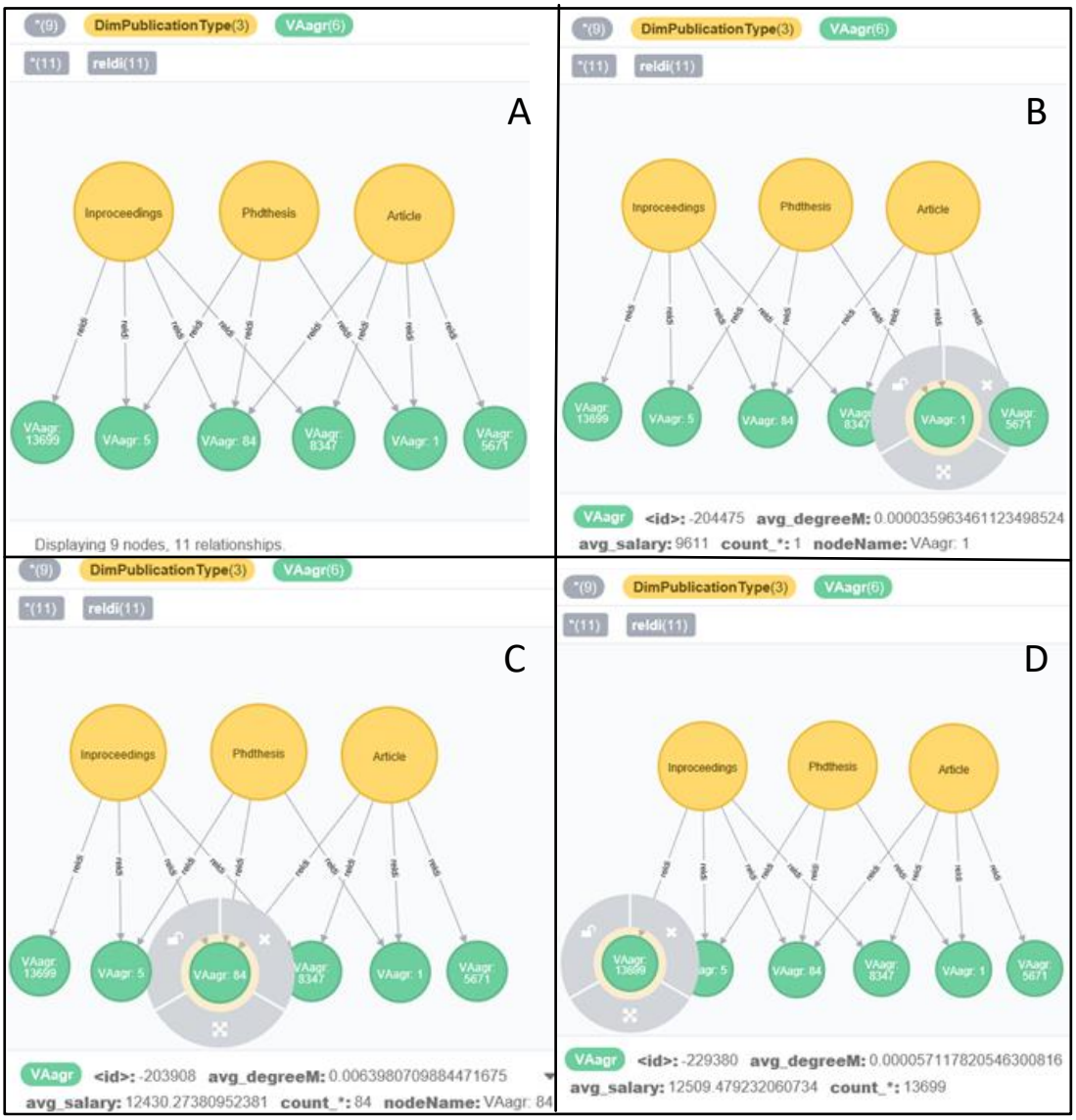


Figura 87 - Resultado da consulta que agrega os autores pelos tipos de publicações.

Elaboramos uma consulta que utiliza duas dimensões {DimDate e DimPublicationType} para analisar os autores que publicaram artigos nos três últimos anos, de modo que os autores que publicaram nos mesmos anos sejam agregados. A Figura 88 apresenta essa consulta especificando na linha 2 a consulta em Cypher que define o

intervalo dos últimos três anos na dimensão DimDate e restringe a dimensão DimPublicationType para o tipo 'Article'; na linha 3 as configurações de análise para os VA; na linha 4 não definimos medidas de análise para os vértices do subgrafo agregado; na linha 5 a configuração dimensional para unificar os VD por meio da configuração {DimDate:'year',DimPublicationType:'type'}; e na linha 6 definimos em Cypher uma adaptação para atender um limitação visual do Neo4j em permitir respostas com até 300 vértices.

```
1  Call olap.aggregate.graphResult(
2  "MATCH (dim_publ:DimPublicationType)-[reldi_publ]->(va:VA),(dim_date:DimDate)-[reldi_date]->(va:VA),(va:VA)-[relSub:relsub]->(subG) where dim_publ.type = 'Article' and dim_date.year >= 2017 and dim_date.year < 2020 RETURN reldi_date,reldi_publ, va, relSub;",
3  [{`*`:'count',salary:['avg'],degreeM:['avg']}],
4  [],
5  {DimDate:'year',DimPublicationType:'type'},
6  ,{graphAggregate:false})YIELD dimensionalNodes,AggregEventNodes,relationships
```

Figura 88 - Consulta que agrega os autores em função publicações dos últimos 3 anos

Com a execução dessa consulta, obtivemos um resultado gráfico em 6415 ms, o qual está retratado na Figura 89. Esse resultado é representado por um grafo de propriedade contendo: três VD {2017, 2018 e 2019} da dimensão DimDate na cor amarela, um VD {Article} da dimensão DimPublicationType e sete VAagr na cor verde. Na agregação, observamos que 972 autores publicaram artigos apenas em 2019; 464 autores publicaram artigos apenas nos anos de 2018 e 2019; e 867 autores publicaram artigos nos três anos consecutivos {2017, 2018 e 2019}.

Considerando que o grafo de resposta apresenta os resultados das medidas de análise nas propriedades dos vértices, apresentamos na Figura 89 quatro imagens com os valores resultantes das medidas de análise. Essas medidas são as mesmas utilizadas na consulta anterior, tendo como diferenciação a especificação das dimensões.

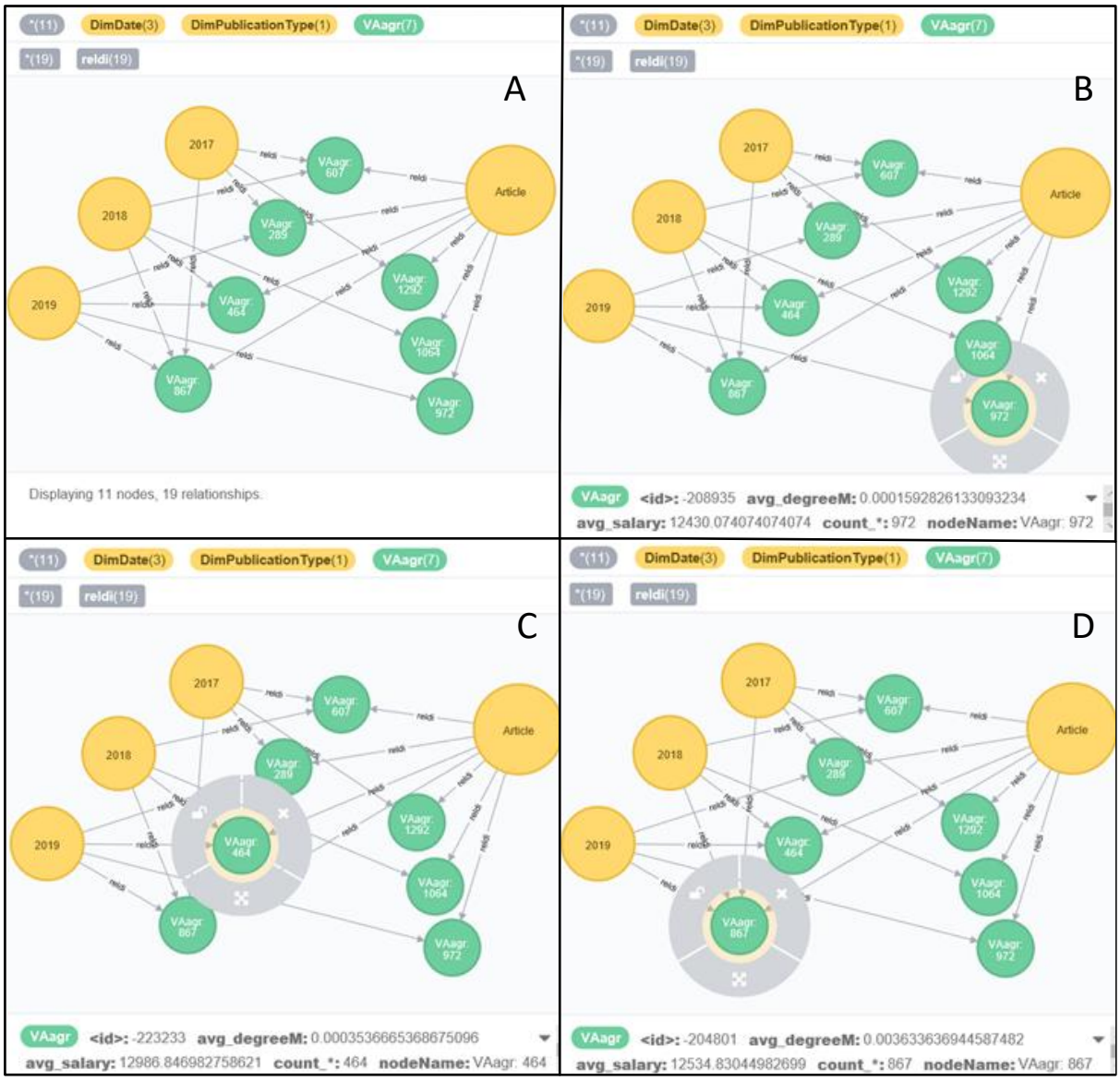


Figura 89 - Resultado da consulta que agrega os autores que publicaram artigos nos últimos 3 anos

Durante a experimentação, observamos que as consultas, as quais solicitavam algum tipo de análise topológica, não conseguiam concluir o processamento em tempo hábil. Além do longo tempo de processamento, as consultas apresentavam resultados de difícil entendimento, apresentado uma grande quantidade de arestas no grafo de resposta.

A Figura 90 é um exemplo de grafo de resposta confuso que contém uma grande quantidade de arestas. Esse grafo de resposta é o resultado da consulta da Figura 88 acrescida com os relacionamentos analíticos (relan) entre os VA. Como podemos observar, a grande quantidade de arestas dificulta a compreensão do resultado e inviabiliza a extração de conhecimento sobre as arestas. Além disso, com o acréscimo dos relan na consulta, o processamento da consulta passou a demorar 5168721 ms, ou seja, aumentou muito o tempo de processamento ao ser comparado com o tempo de 6415 ms da consulta

sem os relan. Com isso, constatamos que o processamento topológico em grande grafo de dados é inviável na AAMPGrafo, pois necessita de um alto custo de processamento para produzir um resultado.

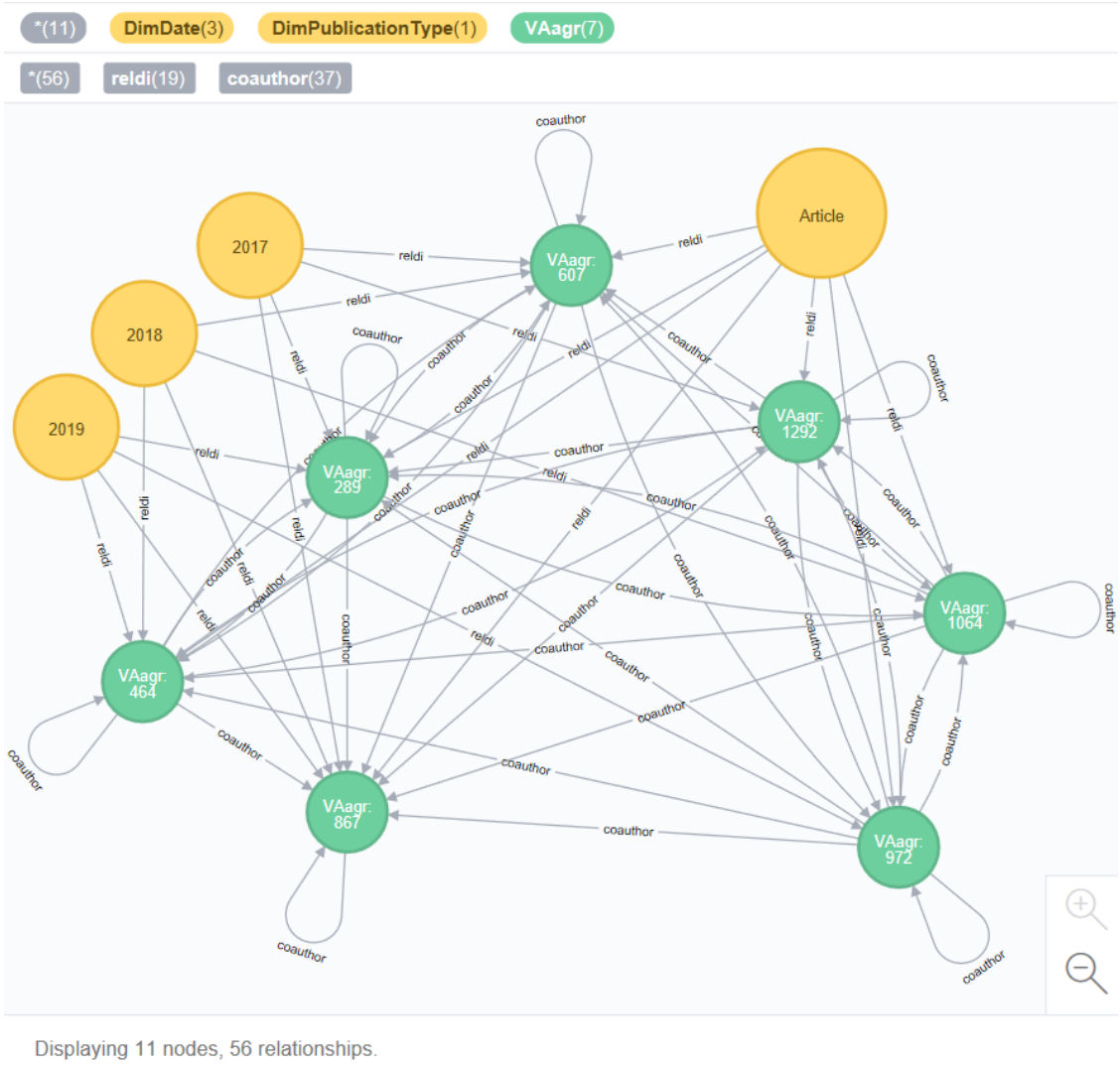


Figura 90 - Resultado da consulta que apresenta as agregações dos autores, que publicaram artigos nos últimos 3 anos, e os seus relacionamentos de coautor

Com base no experimento, constatamos que o volume de dados afeta o processamento e a visualização das consultas multidimensionais em grafo. O processamento topológico do grafo é muito custoso e pode ser impraticável durante a excussão da consulta em uma base volumosa de grafos, principalmente, ao juntar na consulta com outras formas de análise. Com isso, a AAMPGrafo precisa ser repensada para lidar com a topologia de um grande grafo de dados, podendo adotar novas estratégias para melhorar o tempo de resposta nas consultas.

Outra deficiência, é a forma de visualização adotada para lidar com um grande volume de dados em grafo, pois os resultados apresentados são confusos devido à grande quantidade

de arestas. Além disso, a interface gráfica do Neo4j é limitada, permitindo visualizar até 300 vértices, de modo que impossibilita a representação de grades grafos.

Apesar dessa limitações, a AAMPGrafo conseguiu unificar na mesma consulta diferentes formas de análise utilizando grafos de dados de menor volume, como retratamos na exemplificação das consultas. Na próxima seção, apresentamos uma análise comparativa dos trabalhos relacionados enfatizando as formas de se analisar os dados em grafo.

6.5 ANÁLISE COMPARATIVA ENTRE A AAMPGRAFO E OS TRABALHOS RELACIONADOS

Nesta seção, apresentamos uma análise comparativa entre a AAMPGrafo e os trabalhos discutidos no Capítulo 3 (Graph OLAP, Graph Cube, Framework OLAP Cubes, Iceberg Cube e EvOLAP Graph). Na discussão do Capítulo 3, constatamos que essas abordagens possuem o mesmo objetivo: analisar grafos de dados utilizando as tecnologias OLAP. No entanto, para alcançar esse objetivo propuseram diferentes formas de análise. No Capítulo 3, definimos características para compreender e comparar essas abordagens. A Quadro 4 apresenta uma extensão da Quadro 3, inserindo uma nova linha com as características da AAMPGrafo.

Quadro 4 - Comparação da AAMPGrafo com os trabalhos relacionados

Trabalho	Análise top.	Análise OLAP	Análise top. OLAP	Análise de padrões	Grafo agregado	SGBDG	Materialização	Atualização
Graph OLAP	⚠	✓	✗	✗	⚠	✗	⚠	✗
Graph Cube	✗	✓	✗	✗	✓	✗	✓	✗
Framework OLAP Cubes	✓	✓	✓	✗	✓	✓	✓	✗
Iceberg Cube	✗	⚠	✗	✓	⚠	✗	✗	✓
EvOLAP Graph	⚠	⚠	✗	✗	✗	✓	✗	✓
AAMPGrafo	✓	✓	✓	✓	✓	✓	✗	⚠

Na apresentação das consultas é possível constatar a cobertura da AAMPGrafo na maioria das características que descrevem uma forma de análise dos dados ficando com a característica de Atualização com uma avaliação parcial e não contemplando a característica de Materialização de dados, conforme aparece na Quadro 4. Na avaliação dessas características, consideramos que as abordagens cujas consultas são pré-

processadas não são adaptáveis às mudanças, uma vez que ao incluir novos dados é necessário pré-processar na totalidade o grafo de dados.

Considerando esse critério de avaliação, acreditamos que as abordagens EvOLAP Graph e Iceberg Cube atendem à característica de adaptação a mudanças. A abordagem EvOLAP Graph não necessita de pré-processamento para a realização de consultas, mas precisa modificar a estrutura original dos dados para conduzir o controle de versões. O Iceberg Cube também realiza as consultas diretamente no grafo de dados sem precisar de pré-processamento, tendo apenas que especificar os *meta-path* a serem analisados na consulta. Todavia as outras abordagens (Graph OLAP, Graph Cube, Framework OLAP Cubes) realizam o pré-processamento de consulta e, consequentemente, não atendem à característica de adaptação a mudanças. Contudo, essas abordagens realizam a materialização de dados que permite respostas rápidas atendendo melhor as bases de dados de grande volume.

A avaliação da AAMPGrafo é consequência da estratégia de modelar os dados para possibilitar as análises, pois mesmo que a inserção de novos dados seja possível no Modelo MPGrafo, os dados do grafo precisam ser modelados antes da realização da consulta. Dessa forma, consideramos que a AAMPGrafo atende parcialmente à característica de atualização. Além disso, observamos no experimento que apesar de não possuir materialização é possível realizar consultas agregadas em grande volume de dados ao desconsiderar as características topológicas dos dados no processamento da consulta.

Com base nessa análise comparativa, consideramos que a AAMPGrafo cobre mais recursos para a realização de análise em grafo, pois atende, mesmo que de forma parcial, a maioria das características de análise. Essa abordagem permite combinar diferentes formas de análise sobre os mesmos dados, podendo analisar a topologia do grafo e agregar o valor da resultante dessa análise. Apesar das limitações em processar grande volume de dados, a AAMPGrafo expande as formas de análise permitindo ter análises mais específicas em grafo de dados. Além disso, a AAMPGrafo não utiliza estratégias de materialização ou de distribuição para lidar com o processamento de grande volume de dados, podendo utilizar essas estratégias, em trabalhos futuros, para sanar as limitações de processamento topológica em grande volume de dados.

6.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, apresentamos a implementação da AAMPGrafo no SGBDG Neo4j, especificando o *plugin* e os parâmetros das consultas. O *plugin* desenvolvido integra ao SGBDG Neo4j uma funcionalidade que permite consultas multidimensionais em padrões de grafo. Essas consultas utilizam parâmetros para especificar e combinar as formas de análise no grafo de dados. Mostramos neste capítulo as consultas e seus resultados na implementação da AAMPGrafo no Neo4j. Apresentamos um experimento que utiliza um grande grafo de dados para explorar o comportamento da AAMPGrafo em um cenário com grande volume de dados. Discutimos sobre a representação gráfica dos resultados das consultas, realizando uma comparação com as representações encontradas na literatura. Por fim, realizamos uma análise comparativa dos trabalhos relacionados com a AAMPGrafo que desenvolvemos neste trabalho. No capítulo seguinte, apresentamos nossas conclusões e sugestões para trabalhos futuros.

7 CONCLUSÃO E TRABALHOS FUTUROS

A maioria dos dados no mundo real está interconectada, formando redes de informação complexas e heterogêneas (HAN et al., 2012). A análise desses dados tem levado ao desenvolvimento de soluções que integram a tecnologia OLAP e os algoritmos de análise em rede. Essas soluções apresentam diferentes métodos de análise que visam analisar as informações da estrutura topológica e dos valores de propriedade do grafo de dados. Diante desse cenário, o objetivo deste trabalho consistiu em atender as seguintes questões de pesquisa: (Q1) Como integrar algoritmos de análise de rede e tecnologias OLAP a um SGBDG, de modo que permita realizar consultas multidimensionais em padrões de grafo? (Q2) Como representar visualmente os resultados das consultas multidimensionais em padrões de grafo? Com isso, definimos uma hipótese que incentivou a criação de modelos para auxiliar o SGBDG a realizar consultas que combinem análises topológicas e analíticas em padrões de grafo.

Nesse contexto, com vistas a comtemplar essa hipótese, foi desenvolvido uma abordagem de análise que responde a essas questões de pesquisa e atende a hipótese definida. Além disso, corrobora com o campo de Grafo BI, pois auxilia a tomada de decisão combinando diferentes formas de análise para a realização de consultas em redes de informação.

A abordagem, denominada AAMPGrafo, foi definida para ser usada junta a um SGBDG, de modo que habilite funcionalidades para a execução de consultas multidimensionais em padrões de grafo, permitindo tanto análises topológicas e analíticas quanto representações visuais das consultas no SGBDG. Nesta tese, mostramos a implementação de uma instância dessa abordagem no SGBDG Neo4j para exemplificar as formas de análise utilizando a AAMPGrafo. Nessa implementação, desenvolvemos um *plugin* para permitir consultas multidimensionais em padrões de grafo no Neo4j. Os resultados das consultas são expressos na interface gráfica do Neo4j, retratando os valores agregados das medidas de análise e a representação de grafos agregados dentro de uma estrutura orientada aos assuntos das dimensões.

Na avaliação dessa abordagem, selecionamos na revisão sistemática os trabalhos que caracterizam as principais abordagens da área, no intuito de comparar com as características e as funcionalidades desenvolvidas em nosso trabalho. Além disso, destacamos a representação visual das consultas multidimensionais em grafo, com o intuito de trazer uma nova concepção para retratar a análise de dados em redes de informação.

Ainda na avaliação da abordagem, realizamos experimentos para analisar o comportamento da ferramenta ao processar consultas sobre uma base de dados com grande volume de grafos.

Este capítulo é organizado em três seções. A Seção 7.1 discute as contribuições provenientes deste trabalho. A Seção 7.2 indica algumas deficiência e limitações encontradas na abordagem. Seção 7.3 indica algumas direções pelas quais este trabalho pode ser evoluído. A Seção 7.4 conclui a tese com algumas observações finais.

7.1 CONTRIBUIÇÕES DA TESE

As principais contribuições deste trabalho são resumidas a seguir.

I. A definição de um Modelo Multidimensional em padrões de grafo (Modelo MPGrafo), que modela o grafo de dados em uma estrutura multidimensional.

Utilizamos o modelo multidimensional, o modelo de grafo de propriedade em hipervértices e o modelo de grafo de propriedade como conhecimento prévio para definir um modelo multidimensional em grafo de propriedade que permita manipular e analisar padrões de grafo. O padrão de grafo representa uma informação complexa que é expressa por um esquema em grafo (meta grafo). Com isso, o intuito desse modelo é permitir a definição de modelagens orientadas à dimensão para estruturar os padrões de grafo em razão dos assuntos das dimensões, promovendo assim análise e seleção com base nas dimensões especificadas nas consultas. A especificação de um modelo multidimensional em grafo, até onde sabemos, não foi encontrada em trabalhos da literatura.

II. A definição de um Modelo de Consulta Multidimensional em padrões de grafo (Modelo CMPGrafo), que permite análises topológicas e analíticas em consultas multidimensionais em grafo.

Por meio do conhecimento prévio da tecnologia OLAP, dos algoritmos de análise em rede e da álgebra regular de grafo de propriedade "*Regular Property Graph Algebra"* (RPGA) (BONIFATI et al., 2018), definimos um modelo de consulta para atender ao modelo multidimensional em padrões de grafo. Esse modelo possui três etapas de processamento que permitem realizar de forma separada: a recuperação dos dados no SGBDG, a execução de algoritmos de análise em rede, e o processamento de agregação em medidas de análise e topologia do grafo de dados. A divisão dessas etapas facilita a adaptação a outros SGBDG e a aplicação

de outros algoritmos de análise em rede. Até onde pesquisamos na revisão sistemática, a realização de consulta multidimensional em SGBDG é uma abordagem nova que não tinha sido especificada nos trabalhos da literatura.

III. A especificação e implementação de um *plugin* para inserir funcionalidades em um SGBDG que permita a realização de consultas multidimensionais em padrões de grafo com o processamento de medidas de análise topológica e analítica.

Para essa contribuição, usamos os modelos MPGrafo e CMPGrafo com base na implementação de um *plugin* no SGBDG Neo4j. Esse *plugin* integra funcionalidades ao Neo4j para permitir consultas multidimensionais em grafo e proporcionar representações visuais das respostas na interface gráfica do Neo4j. A adaptação do modelo CMPGrafo ao Neo4j resultou na definição de uma função que utiliza parâmetros de configuração e consultas em Cypher para a composição de consultas multidimensionais em padrões de grafo. Com isso, validamos os modelos MPGrafo e CMPGrafo por meio dessa implementação, a qual foi utilizada para apresentar diferentes formas de análises que aplica tecnologia OLAP e algoritmos de análise em rede em padrões de redes de informação.

IV. A implementação de uma representação visual que retrate os resultados das consultas multidimensionais em padrões de grafo, de forma que exiba a agregação topológica do grafo e os valores das medidas de análise.

Considerando as características do grafo de propriedades, destacamos a importância de duas informações: as provenientes das propriedades contidas nos elementos do grafo e as informações sobre a estrutura topológica do grafo (tais como vértices, arestas e rótulos). Com isso, definimos uma representação visual que evidencia a agregação das propriedades do grafo em medidas de análise e a agregação da estrutura topológica dos padrões de grafo. Além disso, essas duas formas de agregação são contextualizadas com informações das dimensões. Na representação visual essa forma de unificar os tipos de agregação de dados consiste em uma nova maneira de retratar a análise de dados ao expor resultados de consultas multidimensionais em padrões de grafo.

7.2 DEFICIÊNCIAS E LIMITAÇÕES

As principais deficiências deste trabalho são resumidas a seguir.

I. Não realizamos uma avaliação para determinar o quão difícil seria para um usuário usar essa ferramenta.

Devido às várias definições apresentadas e a carência de uma interface para o tratamento dos dados é possível que o usuário tenha dificuldade para modelar e consultar os dados. Dessa forma, não sabemos o quão difícil é a aplicação da AAMPGrafo.

II. Há deficiência no processamento da consulta ao requerer análises topológicas em grandes grafos dados.

Durante a experimentação da abordagem utilizando uma base com grandes grafos de dados, observamos que o tempo de resposta das consultas, as quais requeiram algum tipo de análise topológica, é bastante expressivo, podendo demorar mais do que 5168721 ms. Dessa forma, tornasse inviável executar consultas, que combinem medidas topológicas e analíticas, sobre uma base com grandes grafos de dados.

III. Há deficiência na representação visual dos resultados ao realizar consultas em grandes grafos de dados.

Outra deficiência encontrada nos experimentos, é a forma de visualização da consulta, pois os resultados das consultas, quando submetidas a um grande grafo de dados, apresentam um grande volume de arestas. Desse modo, as respostas se tornam confusas e difíceis de compreender. Além disso, durante o experimento foi constatado que a interface gráfica do Neo4j é limitada a reproduzir até 300 vértices. Essa limitação impossibilita representar grandes grafos.

7.3 TRABALHOS FUTUROS

A partir da perspectiva apresentada, destacamos algumas sugestões de melhorias e de novas pesquisas, apontando como possíveis temas para pesquisas futuras, tais como:

- Linguagem de consulta multidimensional em SGBDG

Neste trabalho utilizamos uma função parametrizada para a realização de consultas multidimensionais em padrões de grafo. Essa função poderia ser substituída por uma linguagem de consulta multidimensionais em grafo, tal como a linguagem MDX[18] (Multidimensional Expressions) da Microsoft.

- Ferramenta de ETL para o Modelo MPGrafo

[18] https://docs.microsoft.com/en-us/sql/mdx

A necessidade da AAMPGrafo em modelar o grafo de dados para a realização de consultas pode ser atendida com o desenvolvimento de uma ferramenta para tratar e carregar os dados no formato do Modelo MPGrafo.

- Aplicação da abordagem em um ambiente distribuído

Alguns SGBDG oferecem suporte a ambientes alocados em nuvem, possibilitando a aplicação distribuída da AAMPGrafo. Além disso, a distribuição do processamento é uma possível solução para melhora o processamento topológico durante a consulta.

- Aplicação de diferentes algoritmos de análise em rede

Nesta implementação abordamos as medidas de centralidade em grafo, mas poderiam ser aplicados outros algoritmos de análise em rede e de mineração.

- Análise do desempenho da AAMPGrafo em grandes volumes de dados

Neste trabalho, não realizamos um estudo aprofundado de desempenho para avaliar diferentes modelagens com diferentes volumes de dados.

- Aplicação do Cubo de dados no Modelo MPGrafo

Durante o desenvolvimento desta pesquisa percebemos que a aplicação do Cubo precisaria definir um modelo de cubo em grafo para estruturar os dados pré-processados e um algoritmo de carga para pré-processar e estruturar as possibilidades de consulta nesse modelo de cubo em grafo. Além disso, precisaria de uma linguagem de consulta específica para realizar as consultas.

- Definição de um Modelo para permitir análises multidimensionais em séries temporais em grafo

Com base nas contribuições desta tese, consideramos a possibilidade de representar uma série temporal no formato de um padrão de grafo. Com isso, tornaria possível analisar séries temporais por meio de consultas multidimensionais em padrões de grafo.

7.4 OBSERVAÇÕES FINAIS

Este trabalho investigou a aplicação da tecnológica OLAP em redes de informação, descobrindo um recente campo de pesquisa que realiza agregações de dados em grafo. A AAMPGrafo, fundamentada pela formalização de suas definições básicas, implementações e experimentos realizados, foi apresentada como uma solução para esse cenário. Essa abordagem atende às características das redes de informação heterogêneas de modo que

permita a combinação de análises topológicas e analíticas sobre padrões em grafo de dados. Ademais, estabelece uma representação visual para retratar as respostas das consultas multidimensionais em padrões de grafo. Por conseguinte, mostramos uma solução de análise multidimensionais em padrões de grafo, que permite a representação visual de grafo agregado e de medidas de análise.

REFERÊNCIAS

ANGLES, Renzo. A Comparison of Current Graph Database Models. Apr. 2012, [S.l.]: IEEE, Apr. 2012. p. 171–177. Disponível em: <http://ieeexplore.ieee.org/document/6313676/>. Acesso em: 13 mar. 2017.

ANGLES, Renzo et al. Foundations of Modern Query Languages for Graph Databases. ACM Computing Surveys, v. 50, n. 5, p. 1–40, 26 Sep. 2017. Disponível em: <http://dl.acm.org/citation.cfm?doid=3145473.3104031>. Acesso em: 14 jun. 2019.

BACHMAN, Michal. GraphAware: Towards Online Analytical Processing in Graph Databases. 2013. Disponível em: <http://graphaware.com/assets/bachman-msc-thesis.pdf>. Acesso em: 9 mar. 2017.

BONDY, J. A. e MURTY, U. S. R. Graph theory. [S.l.]: Springer, 2010.

BONDY, J. A. e MURTY, U. S. R. GRAPH THEORY WITH APPLICATIONS. Ontario, Canada: [s.n.], 1976. Disponível em: <http://www.zib.de/groetschel/teaching/WS1314/BondyMurtyGTWA.pdf>. Acesso em: 20 jul. 2019.

BONIFATI, Angela et al. Querying Graphs. Synthesis Lectures on Data Management, v. 10, n. 3, p. 1–184, Oct. 2018. Disponível em: <https://www.morganclaypool.com/doi/10.2200/S00873ED1V01Y201808DTM051>.

BORGATTI, Stephen P. e EVERETT, Martin G. A Graph-theoretic perspective on centrality. Social Networks, v. 28, n. 4, p. 466–484, Oct. 2006.

BRANDES, Ulrik e ERLEBACH, Thomas (Eds.). Network Analysis (Methodological Foundations). Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. v. 3418.

CASTELLTORT, A e LAURENT, A. NoSQL graph-based OLAP analysis. KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, p. 217–224, 2014. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84909972139&partnerID=40&md5=dfb9a731b336cb797037fbb8fb9eb1e7>.

CELKO, Joe. Graph Databases. [S.l: s.n.], 2014. Disponível em: <http://dx.doi.org/10.1016/B978-0-12-407192-6.00003-0>.

CHAUDHURI, Surajit et al. Building the Data Warehouse. Cambridge: Cambridge University Press, 2005. v. 13. Disponível em: <http://190112.8m.com/Bibliografia.pdf%5Cnhttp://doi.wiley.com/10.1002/9780470634431%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.9494&rep=rep1&type=pdf>.

CHEN, Chen et al. Graph OLAP: a multi-dimensional framework for graph data analysis. Knowledge and Information Systems, DBLP, v. 21, n. 1, p. 41–63, 1 Oct. 2009. Disponível em: <http://link.springer.com/10.1007/s10115-009-0228-9>.

CHEN, Chen et al. Graph OLAP: Towards Online Analytical Processing on Graphs. Dec.

2008, [S.l.]: IEEE, Dec. 2008. p. 103–112. Disponível em: <http://ieeexplore.ieee.org/document/4781105/>.

CYGANIAK, Richard e WOOD, David e LANTHALER, Markus. RDF 1.1 Concepts and Abstract Syntax. Disponível em: <https://www.w3.org/TR/rdf11-concepts/>. Acesso em: 25 jul. 2019.

DENIS, Benoit e GHRAB, Amine e SKHIRI, Sabri. A distributed approach for graph-oriented multidimensional analysis. Oct. 2013, [S.l.]: IEEE, Oct. 2013. p. 9–16. Disponível em: <http://ieeexplore.ieee.org/document/6691777/>.

DIESTEL, Reinhard. Graph Theory (Graduate Texts in Mathematics). [S.l.]: Springer, 2005.

FANG, Min et al. Computing Iceberg Queries Efficiently*. 1998, New York, USA,: [s.n.], 1998. Disponível em: <http://www.vldb.org/conf/1998/p299.pdf>. Acesso em: 4 jan. 2018.

FREEMAN, Linton C. Centrality in social networks conceptual clarification. Social Networks, v. 1, n. 3, p. 215–239, Jan. 1978. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0378873378900217>.

GHRAB, Amine et al. A Framework for Building OLAP Cubes on Graphs. Advances in Databases and Information Systems. [S.l: s.n.], 2015. p. 92–105. Disponível em: <http://code.ulb.ac.be/dbfiles/GhrRomSkh-etal2015incollection.pdf>. Acesso em: 4 may 2017.

GHRAB, Amine et al. An Analytics-Aware Conceptual Model for Evolving Graphs. Data Warehousing and Knowledge Discovery. [S.l: s.n.], 2013. p. 1–12. Disponível em: <http://link.springer.com/10.1007/978-3-642-40131-2_1>. Acesso em: 13 dec. 2016.

GHRAB, Amine et al. Analytics-Aware Graph Database Modeling. . [S.l: s.n.], 2014. Disponível em: <https://research.euranova.eu/wp-content/uploads/analytics-aware-graph-database-modeling.pdf>. Acesso em: 11 mar. 2017.

GHRAB, Amine et al. Graph BI & Analytics: Current State and Future Challenges. [S.l: s.n.], 2018. p. 3–18. Disponível em: <http://link.springer.com/10.1007/978-3-319-98539-8_1>. Acesso em: 5 jul. 2019.

GORUNESCU, Florin. Data Mining: Concepts and Techniques. [S.l: s.n.], 2011. v. 12. Disponível em: <http://link.springer.com/10.1007/978-3-642-19721-5>.

GUMINSKA, Ewa e ZAWADZKA, Teresa. EvOLAP Graph – Evolution and OLAP-Aware Graph Data Model. [S.l: s.n.], 2018. p. 75–89. Disponível em: <http://link.springer.com/10.1007/978-3-319-99987-6_6>. Acesso em: 5 jul. 2019.

HAN, Jiawei et al. Mining Knowledge from Data: An Information Network Analysis Approach. 2012 IEEE 28th International Conference on Data Engineering, p. 1214–1217, 2012. Disponível em: <http://ieeexplore.ieee.org/document/6228171/>.

HEUDECKER, Nick e FEINBERG, Donald. IT Market Clock for Database Management Systems, 2014. Disponível em: <https://www.gartner.com/en/documents/2852717>. Acesso em: 5 jul. 2019.

INMON, William H. Building the data warehouse. [S.l.]: Wiley, 1993. Disponível em: <https://dl.acm.org/citation.cfm?id=531510>. Acesso em: 24 jul. 2019.

INMON, William H. Building the data warehouse. [S.l.]: Wiley Pub, 2005.

JAKAWAT, Wararat e FAVRE, Cécile e LOUDCHER, Sabine. Graphs enriched by cubes for OLAP on bibliographic networks. International Journal of Business Intelligence and Data Mining, v. 11, n. 1, p. 85, 2016a. Disponível em: <http://www.inderscience.com/link.php?id=76435>.

JAKAWAT, Wararat e FAVRE, Cécile e LOUDCHER, Sabine. OLAP cube-based graph approach for bibliographic data. 2016b, Harrachov, Czech Republic: [s.n.], 2016. p. 87–99.

JEH, Glen e WIDOM, Jennifer. SimRank: A Measure of Structural-Context Similarity. 2002, New York, USA: ACM Press, 2002. p. 538. Disponível em: <https://cs.ucsb.edu/~victor/pub/ucsb/mae/references/widom-simrank-2002.pdf>. Acesso em: 6 mar. 2017.

JIAWEI, Han e KAMBER, Micheline e PEI, Jian. Data mining. Concepts and Techniques. [S.l: s.n.], 2012.

KIMBALL, Ralph. e ROSS, Margy. The data warehouse toolkit : the complete guide to dimensional modeling. [S.l.]: Wiley, 2002a.

KIMBALL, Ralph e ROSS, Margy. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. [S.l: s.n.], 2002b.

KIMBALL, Ralph e ROSS, Margy. The Data Warehouse Toolkit. [S.l: s.n.], 2013. Disponível em: <http://www.essai.rnu.tn/Ebook/Informatique/The Data Warehouse Toolkit, 3rd Edition.pdf>. Acesso em: 16 jul. 2019.

KITCHENHAM, B. e CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. . Durham, UK: EBSE Technical Report, 2007. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.471>. Acesso em: 22 jun. 2017.

LEE, Sangkeun et al. Enabling graph mining in RDF triplestores using SPARQL for holistic in-situ graph analysis. Expert Systems with Applications, v. 48, p. 9–25, 2016. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2015.11.010>.

LIU, Y e VITOLO, T M. Graph data warehouse: Steps to integrating graph databases into the traditional conceptual structure of a data warehouse. Proceedings - 2013 IEEE International Congress on Big Data, BigData 2013, p. 433–434, 2013. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84886000078&partnerID=40&md5=0b6eb70b65965e19c781c4db51fbcfbc>.

LOUDCHER, Sabine et al. Combining OLAP and information networks for bibliographic data analysis: a survey. Scientometrics, v. 103, n. 2, p. 471–487, 17 May 2015. Disponível em: <http://link.springer.com/10.1007/s11192-015-1539-0>.

MILLER, Justin. Graph Database Applications and Concepts with Neo4j. SAIS 2013Proceedings, 2013. Disponível em: <http://aisel.aisnet.org/sais2013/24>. Acesso em:

13 mar. 2017.

MPINDA, Steve Ataky Tsham e BUNGAMA, Patrick Andjasubu e MASCHIETTO, Luis Gustavo. Graph Database Application using Neo4j (Railroad Planner Simulation). International Journal of Engineering Research and, v. V4, n. 04, 25 Apr. 2015. Disponível em: <http://www.ijert.org/view-pdf/12960/graph-database-application-using-neo4j-railroad-planner-simulation>. Acesso em: 21 feb. 2017.

NAN LI et al. gIceberg: Towards iceberg analysis in large graphs. Apr. 2013, [S.l.]: IEEE, Apr. 2013. p. 1021–1032. Disponível em: <http://ieeexplore.ieee.org/document/6544894/>.

NEWMAN, Mark. Networks: An Introduction. 1. ed. [S.l.]: Oxford University Press, 2010.

NEWMAN, Mark. Networks. [S.l.]: Oxford University Press, 2018. v. 1. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=YdZjDwAAQBAJ&oi=fnd&pg=PP1&dq=Networks:+An+Introduction+2018&ots=V_J-_Nl9ox&sig=aGrnXx4y7kbPasxUy_wMx-2G53Q#v=onepage&q=Networks%3A An Introduction 2018&f=false>. Acesso em: 17 dec. 2018.

OPSAHL, Tore e AGNEESSENS, Filip e SKVORETZ, John. Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, v. 32, n. 3, p. 245–251, 2010.

PAGE, Lawrence et al. The PageRank Citation Ranking: Bringing Order to the Web. . [S.l.]: Stanford InfoLab, Nov. 1999.

PAMPA QUISPE, Newton Roy. Tecnicas e ferramentas para a extração inteligente e automatica de conhecimento em banco de dados. 2003. [s.n.], 2003. Disponível em: <http://repositorio.unicamp.br/jspui/handle/REPOSIP/260072>. Acesso em: 14 mar. 2017.

PANZARINO, Onofrio. Learning Cypher. [S.l.]: Packt Publishing, 2014.

POKORNÝ, Jaroslav. Conceptual and Database Modelling of Graph Databases. 2016, New York, New York, USA: ACM Press, 2016. p. 370–377. Disponível em: <http://dl.acm.org/citation.cfm?doid=2938503.2938547>.

QUEIROZ-SOUSA, Paulo orlando e SALGADO, Ana carolina. A Review on OLAP Technologies Applied to Information Networks. Transactions on Knowledge Discovery from Data, 2019.

RASLAN, Daniela Andrade e CALAZANS, Angélica Toffano Seidel. Data Warehouse: conceitos e aplicações. Universitas: Gestão e TI, v. 4, n. 1, 4 Aug. 2014. Disponível em: <http://www.publicacoes.uniceub.br/index.php/gti/article/view/2612>.

ROZEVA, Anna e ANNA. Dimensional hierarchies. 2007, New York, New York, USA: ACM Press, 2007. p. 1. Disponível em: <http://portal.acm.org/citation.cfm?doid=1330598.1330648>. Acesso em: 14 mar. 2017.

RUOHONEN, Keijo. GRAPH THEORY. . [S.l: s.n.], 2013. Disponível em: <http://math.tut.fi/~ruohonen/GT_English.pdf>. Acesso em: 20 jul. 2019.

SHI, Chuan et al. A Survey of Heterogeneous Information Network Analysis. arXiv preprint

arXiv:1511.04854, v. 14, n. 8, p. 1–45, 2015. Disponível em: <http://arxiv.org/abs/1511.04854>.

SHI, Chuan e YU, Philip S. Heterogeneous Information Network Analysis and Applications. Cham: Springer International Publishing, 2017. Disponível em: <http://link.springer.com/10.1007/978-3-319-56212-4>. (Data Analytics).

SUN, Yizhou e HAN, Jiawei. Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Explorations Newsletter, v. 14, n. 2, p. 20, 30 Apr. 2013. Disponível em: <http://www.kdd.org/exploration_files/V14-02-03-Sun.pdf>. Acesso em: 4 may 2017.

SUN, Yizhou e HAN, Jiawei. Mining Heterogeneous Information Networks: Principles and Methodologies. Synthesis Lectures on Data Mining and Knowledge Discovery, v. 3, n. 2, p. 1–159, 18 Jul. 2012. Disponível em: <http://www.morganclaypool.com/doi/abs/10.2200/S00433ED1V01Y201207DMK005>. Acesso em: 4 may 2017.

VAISMAN, A. Conceptual Modeling of Data Warehouses. [S.l: s.n.], 2014.

VAISMAN, Alejandro e ZIMÁNYI, Esteban. Data Warehouse Systems. [S.l: s.n.], 2014. Disponível em: <http://link.springer.com/10.1007/978-3-642-54655-6>.

VASILAKIS, C. e EL-DARZI, E. e CHOUNTAS, P. A Data Warehouse Environment for Storing and Analyzing Simulation Output Data. 2004, [S.l.]: IEEE, 2004. p. 691–698. Disponível em: <http://ieeexplore.ieee.org/document/1371379/>. Acesso em: 14 mar. 2017.

VIJITBENJARONK, Warut D. et al. Scalable time-versioning support for property graph databases. Dec. 2017, [S.l.]: IEEE, Dec. 2017. p. 1580–1589. Disponível em: <http://ieeexplore.ieee.org/document/8258092/>. Acesso em: 8 jul. 2019.

WANG, Pengsen e WU, Bin e WANG, Bai. TSMH Graph Cube: A novel framework for large scale multi-dimensional network analysis. Oct. 2015, [S.l.]: IEEE, Oct. 2015. p. 1–10. Disponível em: <http://ieeexplore.ieee.org/document/7344826/>.

WANG, Zhengkui et al. Pagrol: Parallel graph olap over large-scale attributed graphs. Mar. 2014, [S.l.]: IEEE, Mar. 2014. p. 496–507.

YIN, Dan et al. Approximate Iceberg Cube on Heterogeneous Dimensions. Database Systems for Advanced Applications. [S.l: s.n.], 2016. v. 9049. p. 82–97.

YIN, Dan e GAO, Hong. Iceberg Cube Query on Heterogeneous Information Networks. Wireless Algorithms, Systems, and Applications. [S.l: s.n.], 2014. p. 740–749.

YIN, Mu e WU, Bin e ZENG, Zengfeng. HMGraph OLAP: a Novel Framework for Multi-dimensional Heterogeneous Network Analysis. 2012, New York, USA: ACM Press, 2012. p. 137.

ZHANG, Zixing e WU, Bin e WANG, Zeao. A Parallel Framework for Large-scale Multidimensional Heterogeneous Network Analysis. 2017, New York, USA: ACM Press, 2017. p. 625–626.

ZHAO, Peixiang et al. Graph Cube: OnWarehousing and OLAP Multidimensional Networks. 2011, New York, USA: ACM Press, 2011. p. 853.

APÊNDICE A - EXEMPLOS DE CONSULTA NO SGBDG

- Modelagem MPGrafo-1

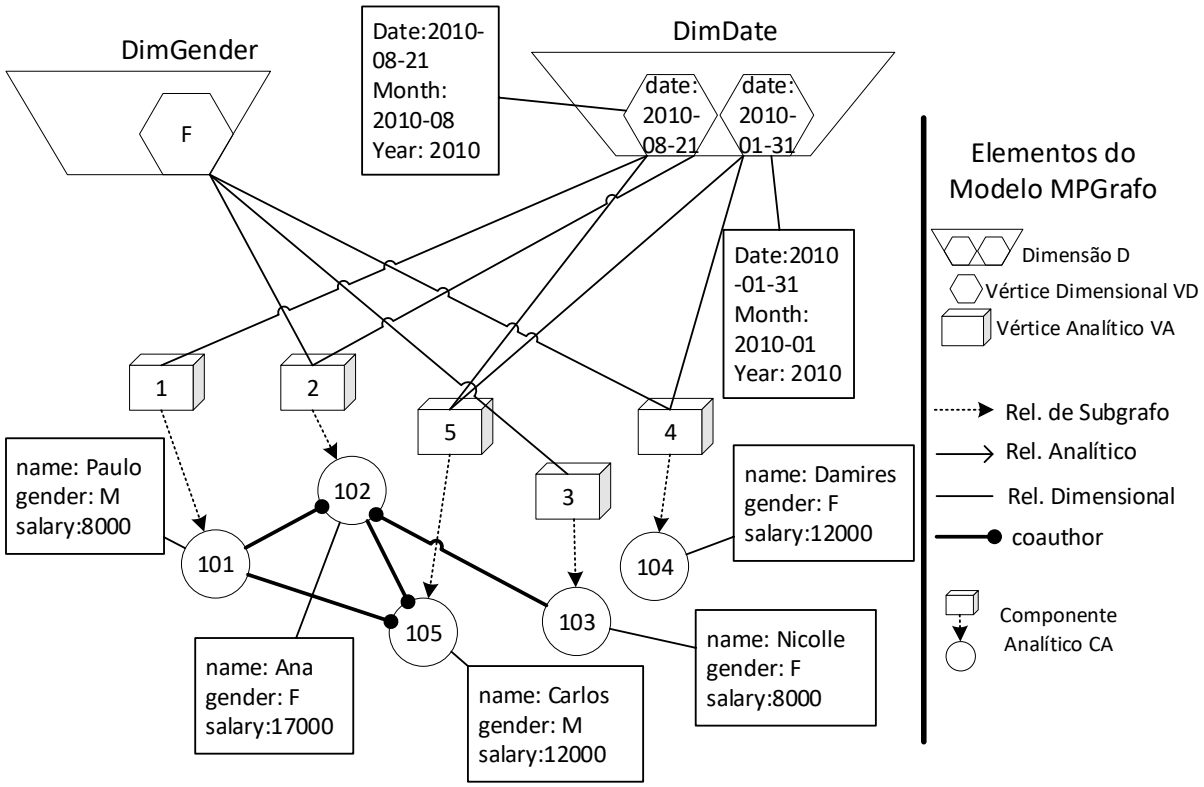


Figura 91 - Consulta os autores que publicaram em 2010 ou são do gênero feminino

A Figura 91 mostra a consulta dos autores que publicaram em 2010 ou são do gênero feminino. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,trg_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{trg_3} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\,(src_1)$ = :DimDate $\wedge\, src_1$.year = 2010 $\wedge\, \lambda\,(trg_1) = r_{VA} \wedge \lambda\,(src_2)$ = :DimGender $\wedge\, src_2$.gender = "F" $\wedge\, \lambda\,(trg_2) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_3 \wedge\ trg_2 = src_3$

Nessa consulta, ϕE_{VA} requisita dois tipos de relacionamentos: o primeiro, entre os vértices src_1 de rótulo "DimDate" com o valor "2010" na propriedade 'year' e os vértices trg_1 de rótulo r_{VA}; e o segundo, entre os vértices src_2 de rótulo ":DimGender" com o valor "F" na propriedade 'gender' e os vértices trg_2 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 e trg_2 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_3 \in subV$.

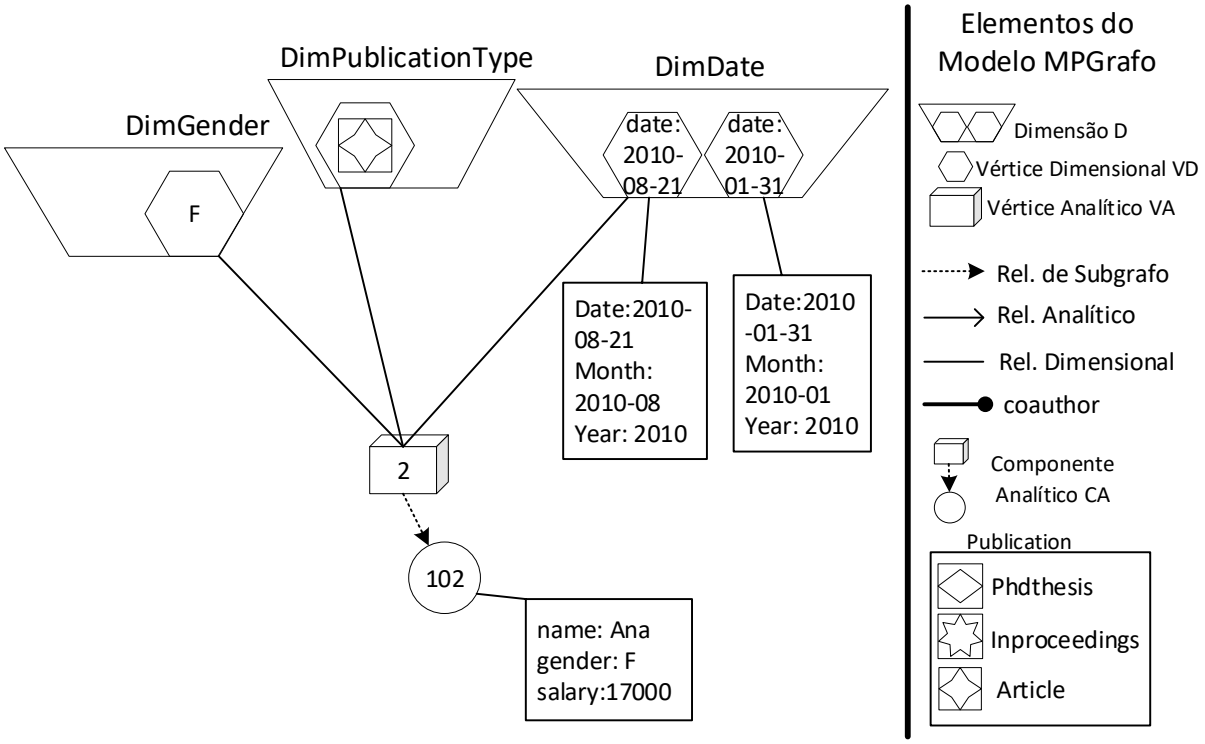


Figura 92 - Consulta os autores que são do gênero feminino e publicaram um artigo em 2010.

A Figura 92 mostra a consulta dos autores que são do gênero feminino e publicaram um artigo em 2010. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,src_2,src_3,trg_3} \left(\bowtie^{\phi E_{sub}}_{trg_4} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda(src_1)$ = :DimDate $\wedge$ src_1.year = 2010 $\wedge$ $\lambda(trg_3) = r_{VA} \wedge \lambda(src_2)$ = :DimGender $\wedge$ src_2.gender = "F" $\wedge$ (src_3) = :DimPublicationType $\wedge$ src_3.type = "Article" $\wedge$ $trg_1 = trg_3 \wedge trg_2 = trg_3$
- ϕE_{sub}: $trg_3 = src_4$

Nessa consulta, ϕE_{VA} requisita três tipos de relacionamentos: o primeiro, entre os vértices src_1 de rótulo "DimDate" com o valor "2010" na propriedade "year" e os vértices trg_1 de rótulo r_{VA}; o segundo, entre os vértices src_2 de rótulo ":DimGender" com o valor "F" na propriedade "gender" e os vértices trg_2 de rótulo r_{VA}; e o terceiro, entre os vértices src_3 de rótulo ":DimPublicationType" com o valor "Article" na propriedade "type" e os vértices trg_3 de rótulo r_{VA}. Além disso, trg_1, trg_2 e trg_3 correspondem aos mesmos vértices, sendo representados por trg_3 na consulta. ϕE_{sub} especifica que os vértices encontrados trg_3 são representados por src_4 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_4 \in subV$.

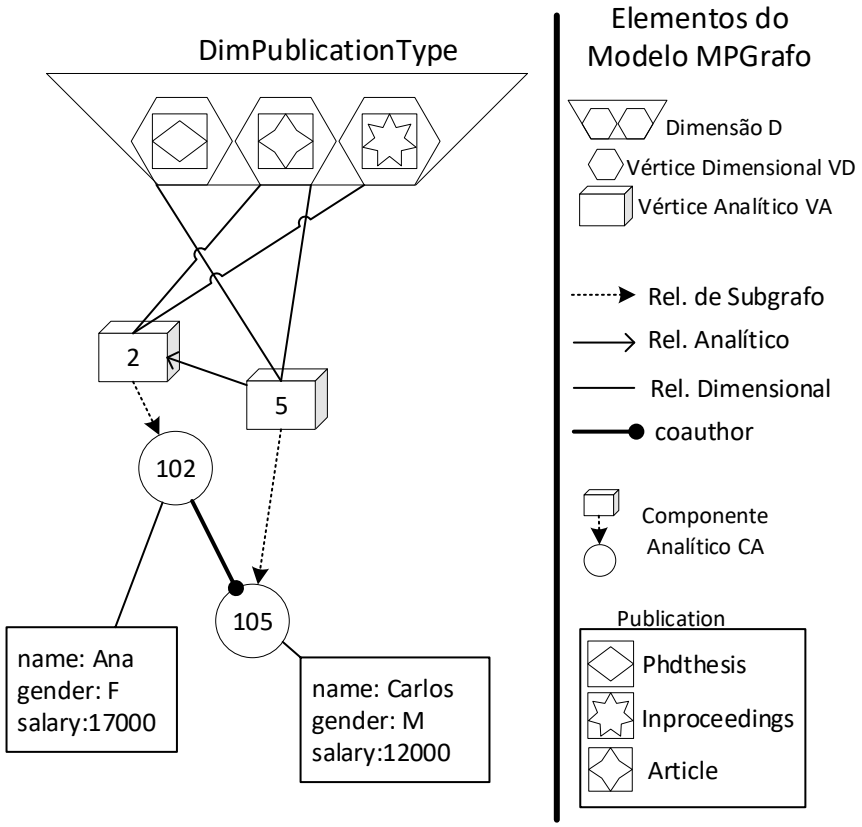


Figura 93 - Consulta os autores que possuem mais de uma publicação

A Figura 93 mostra a consulta dos autores que possuem mais de uma publicação. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,\ trg_1} \left(\bowtie^{\phi E_{sub}}_{src_2,\ trg_2} (r_{relsub})\right)$, , onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimPublicationType $\wedge\ \lambda\ (trg_1)$ = :r_{VA} $\wedge$ COUNT($edge_1$) > 1
- ϕE_{sub}: $trg_1 = src_2 \wedge src_2 = trg_2 \wedge \lambda\ (src_2) = r_{VA}$

Nessa consulta, ϕE_{VA} solicita que a quantidade de relacionamentos $edge_1$, entre os vértices src_1 de rótulo ":DimPublicationType" e os vértices trg_1 de rótulo r_{VA}, seja maior que "1" . ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$ e os vértices src_2 possuem relacionamentos entre si

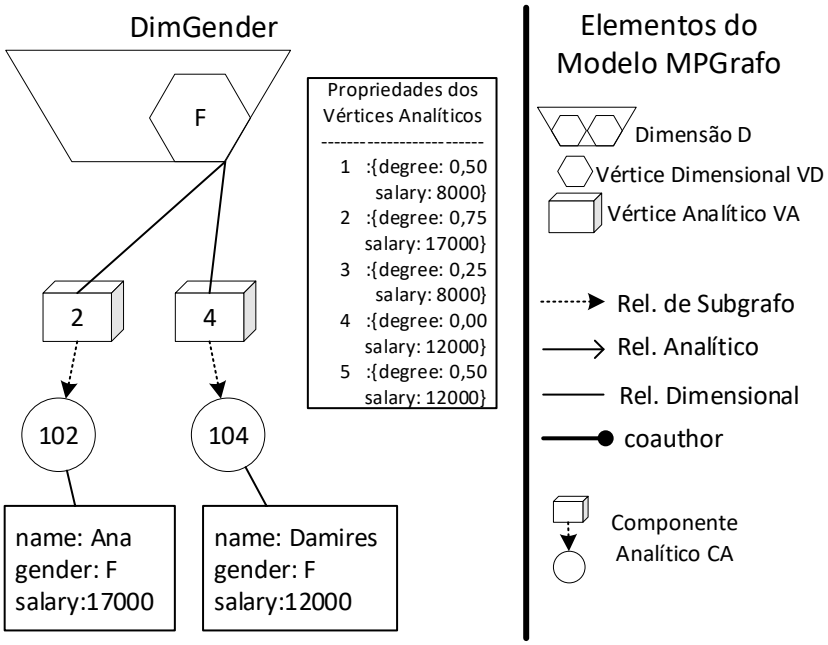


Figura 94 - Consulta os autores do gênero feminino com salário maior que 10000

A Figura 94 mostra a consulta dos autores do gênero feminino com salário maior que 10000. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimGender $\wedge$ src_1.gender = "F" $\wedge$ $\lambda\ (trg_1) = r_{VA} \wedge \ trg_1.salary.>$ 10000
- ϕE_{sub}: $trg_1 = src_2$

Nessa consulta,ϕE_{VA} requisita o relacionamento entre os vértices src_1 de rótulo ":DimGender" com o valor "F" na propriedade 'gender' e os vértices trg_1 de rótulo r_{VA}, que contêm a propriedade "salary" maior que "10000". ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$.

- Modelagem MPGrafo-2

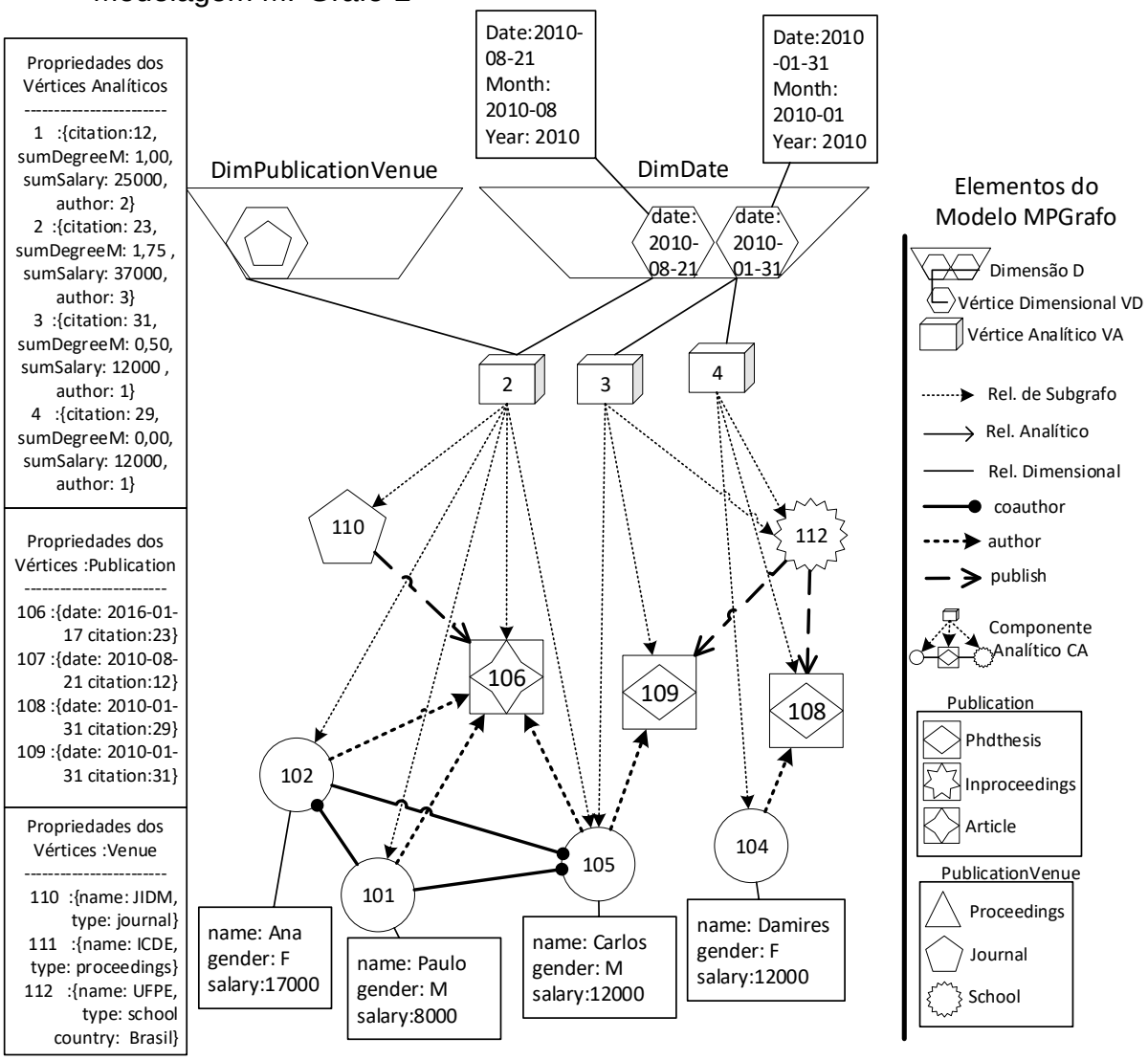


Figura 95 - Consulta as publicações que são de 2010 ou foram publicadas em “Journal”.

A Figura 95 mostra a consulta das publicações que são de 2010 ou foram publicadas em “Journal”. A expressão formal da consulta segue o seguinte formato:

$\bowtie^{\phi E_{VA}}_{src_1,trg_1,src_2,trg_2} \left(\bowtie^{\phi E_{sub}}_{trg_3} (r_{relsub})\right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge$ src_1.year = 2010 $\wedge$ $\lambda\,(trg_1) = r_{VA} \wedge \lambda\ (src_2)$ = :DimPublicationVenue $\wedge$ src_2.type = “Journal” $\wedge\ \lambda\ (trg_2) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_3 \wedge trg_2 = src_3$

Nesse exemplo, ϕE_{VA} requisita dois tipos de relacionamentos: o primeiro, entre os vértices src_1 de rótulo “DimDate” com o valor “2010” na propriedade ‘year’ e os vértices trg_1 de rótulo r_{VA}; e o segundo, entre os vértices src_2 de rótulo “:DimPublicationVenue” com o valor “Journal” na propriedade “type” e os vértices trg_2 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 e trg_2 são representados por src_2 nos relacionamentos de rótulo

r_{relsub} com os vértices $trg_3 \in subV$.

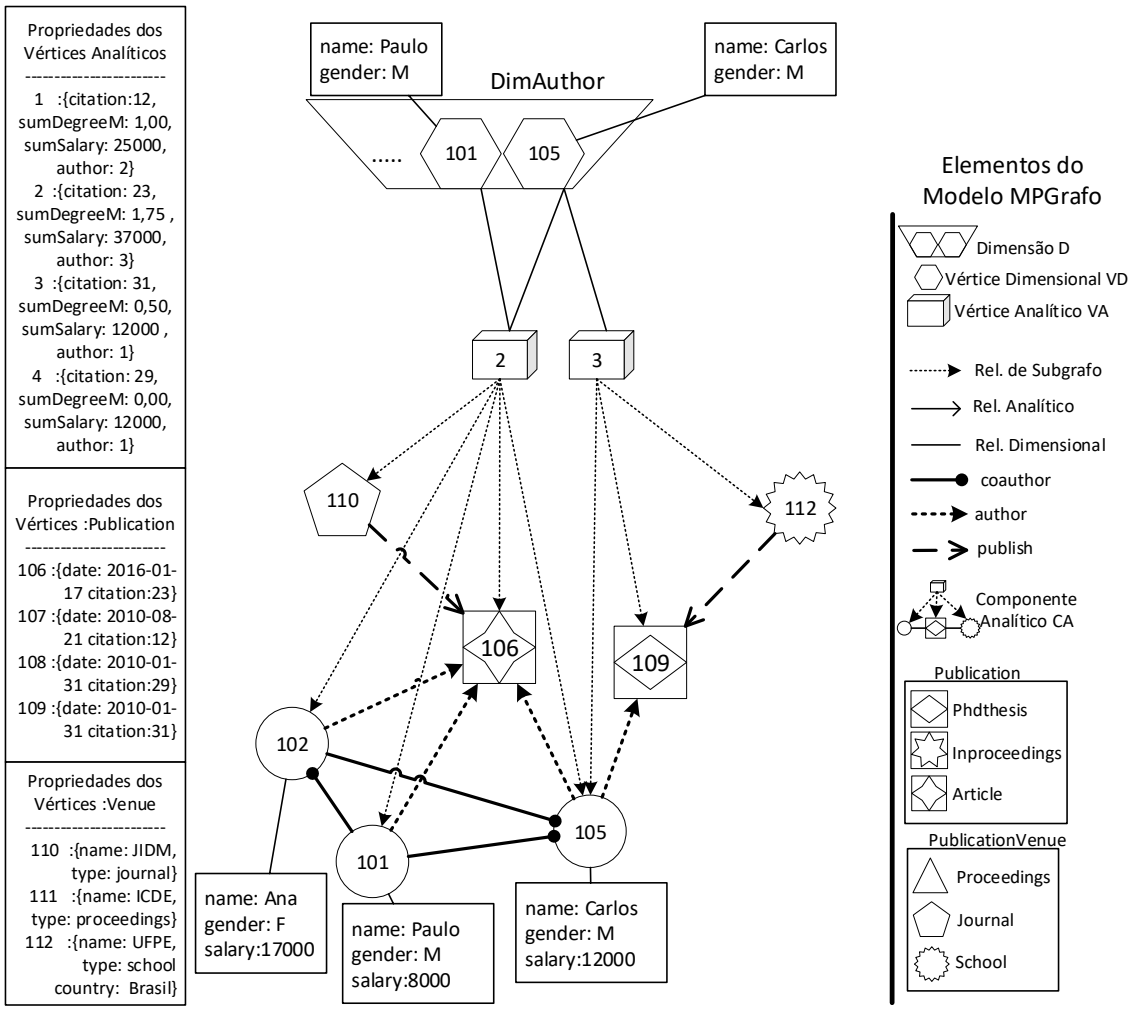


Figura 96 - Consulta as publicações que possuem autores do gênero masculino e mais de 20 citações

A Figura 96 mostra a consulta das publicações que possuem autores do gênero masculino e mais de 20 citações. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimGender $\wedge$ src_1.gender = "M" $\wedge$ $\lambda\ (trg_1) = r_{VA} \wedge \ \ trg_1.citation.>$ 20
- ϕE_{sub}: $\ trg_1 = src_2$

Nessa consulta, ϕE_{VA} requisita o relacionamento entre os vértices src_1 de rótulo ":DimGender" com o valor "M" na propriedade 'gender' e os vértices trg_1 de rótulo r_{VA}, que possui o valor maior que "20" na propriedade "citation". ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$.

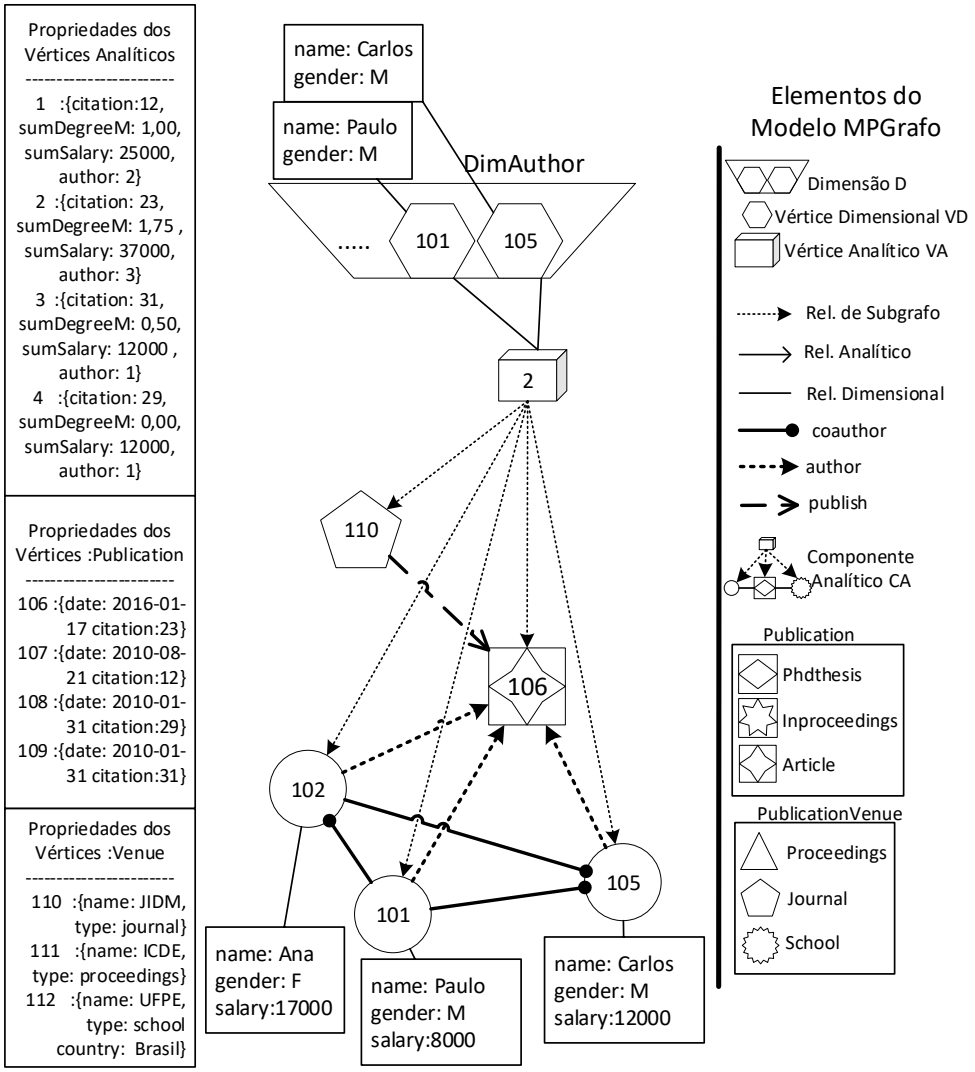


Figura 97 - Consulta as publicações com mais de um autor do gênero masculino

A Figura 97 mostra a consulta das publicações com mais de um autor do gênero masculino. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimAuthor $\wedge$ src_1.gender = "M" $\wedge$ $\lambda\ (trg_1)$ = :r_{VA} $\wedge$ COUNT($edge_1$) > 1
- ϕE_{sub}: $trg_1 = src_2$

Nessa consulta, ϕE_{VA} solicita que a quantidade de relacionamentos $edge_1$ entre os vértices src_1 de rótulo ":DimGender" com o valor "M" na propriedade 'gender' e os vértices trg_1 de rótulo r_{VA}, seja maior que "1". ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$.

APÊNDICE B - EXEMPLO DE CONSULTA COM PARÂMETRO DIMENSIONAL

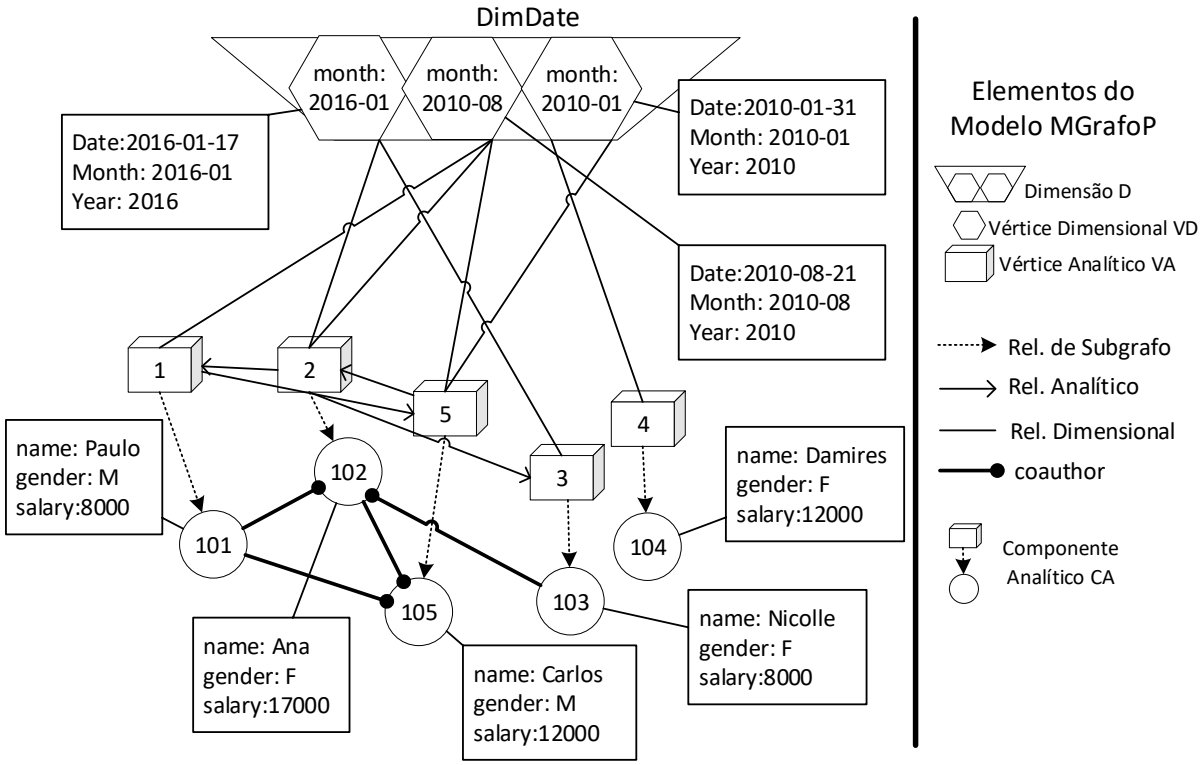


Figura 98 - Consulta e agrega os autores em função dos meses de publicação

A Figura 98 mostra o resultado da consulta que agrega os autores considerando os meses de publicação. A expressão formal da consulta segue o seguinte formato:

$\bowtie_{src_1,trg_1}^{\phi E_{VA}} \left(\bowtie_{trg_2}^{\phi E_{sub}} (r_{relsub}) \right)$, onde

- ϕE_{VA}: $\lambda\ (src_1)$ = :DimDate $\wedge\ \lambda\ (trg_1) = r_{VA}$
- ϕE_{sub}: $trg_1 = src_2$
- Configuração Dimensional: { DimDate:[month] }

Nessa consulta, ϕE_{VA} requisita os relacionamentos entre os vértices src_1 de rótulo ":DimDate" e os vértices trg_1 de rótulo r_{VA}. ϕE_{sub} especifica que os vértices encontrados trg_1 são representados por src_2 nos relacionamentos de rótulo r_{relsub} com os vértices $trg_2 \in subV$. Após a execução dessa consulta, os VD recuperados são unificados, considerando o rótulo "DimDate" e os valores existentes na propriedade "month".