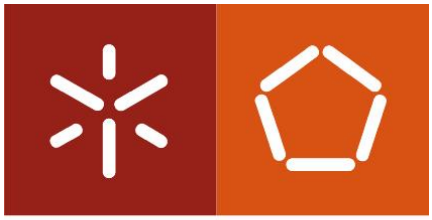
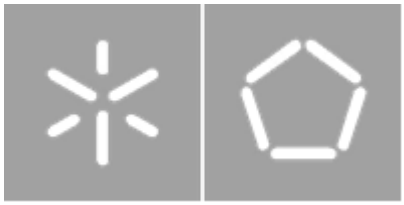


Universidade do Minho
Escola de Engenharia

Carlos Guilherme Marques Gregório

Esfoliação de Hipercubos



Universidade do Minho
Escola de Engenharia
Departamento Informática

Carlos Guilherme Marques Gregório

Esfoliação de Hipercubos

Dissertação de Mestrado

Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação de

Professor Orlando Belo

Agradecimentos

Queria deixar um agradecimento ao meu orientador Orlando Belo pelo auxílio que me prestou ao longo do desenvolvimento desta dissertação.

Um agradecimento à Universidade do Minho, em especial ao Departamento de Informática pelos serviços e contributos prestados para a minha formação.

Por fim e como não poderia deixar de ser, deixo também aqui um agradecimento à minha família e amigos.

Resumo

Um hipercubo é uma estrutura de dados bastante importante no suporte a processos de análise de informação e respectiva tomada de decisão. Contudo nem todos os dados contidos num hipercubo são utilizados nos processos de exploração promovidos pelos agentes de decisão de uma dada organização. Basicamente, isto significa que, por norma, um hipercubo tem mais dados do que aqueles que realmente são necessários na prática, no quotidiano de um agente de decisão. Este trabalho de dissertação visa estudar a redução do tamanho dos hipercubos de uma dada organização e o aumento do desempenho dos processos que sobre eles possam atuar. O estudo a desenvolver passa por uma análise dos resultados das queries mais frequentes lançadas pelos utilizadores de forma a estabelecer um plano de materialização mais eficiente, que apenas considere as partes de um hipercubo que de facto são utilizadas na satisfação regular dos pedidos dos agentes de decisão.

Palavras-chave: Hipercubo, Processos de análise, Tomada de decisão, Agentes de Decisão, *Queries* mais frequentes, Plano de Materialização.

Abstract

A hypercube is a very important data structure in support decision process analysis of information and respective decision making. However not all data contained in a hypercube are used in the exploration processes promoted by a given organization decision makers. Basically, this means that, by rule, a hypercube has more data than those that are really needed in practice by the decision makers. This dissertation aims to study the reduction in size of hypercubes of a given organization and increasing the performance of processes that can act on them. The study to develop requires an analysis of the results of the most frequent queries posted by users in order to establish a more efficient realization plan, which only consider the parts of a hypercube that in fact are relevant to reply decision makers needs.

Keywords: Hypercube, Process Analysis, Decision making, Most Frequent *Queries*, Realization Plan.

Índice

Índice de Figuras

Índice de tabelas

Capítulo 1

Introdução

1.1. Contextualização

Hoje, a informação é particularmente valiosa para qualquer empresa e, como tal, deve ser devidamente guardada e acondicionada em repositórios adequados de forma a estar prontamente acessível na altura em que for requerida. Contudo, a existência de grandes volumes de dados nesses repositórios, muitas vezes descentralizados, torna difícil a extração de conhecimento relevante dos processos operacionais, se não mesmo, por vezes, impossível (Golfarelli & Rizzi 2009). O desafio passa então por transformar esses dados numa fonte de conhecimento, com o objectivo de construir uma vista consolidada da organização. Uma forma de obter esta vista é através do desenvolvimento de um *data warehouse*, que converge dados de múltiplas fontes e os organiza em esquemas multidimensionais de acordo com os requisitos dos agentes de decisão, que permitem depois uma análise dos dados em diferentes perspectivas e níveis detalhe (Connolly & Begg, 2005). Assim é obtida uma correspondência direta entre os requisitos dos agentes de decisão e os dados existentes nas fontes. Então, cria-se a ponte para o uso de ferramentas de análise. As principais, ou pelo menos as mais populares, ferramentas são as *OLAP*, que permitem analisar e explorar dados de um *data warehouse* de forma interativa e fácil (Golfarelli & Rizzi, 2009). Tradicionalmente, as aplicações Online Analytical Processing (OLAP) são baseadas em modelos multidimensionais que, intuitivamente, representam os dados sobre a metáfora de um cubo cujas células

correspondem a eventos que ocorreram no âmbito do negócio (Rizzi, 2009). As ferramentas OLAP são muitas vezes uma solução para o tipo dos processos de análise que referimos, propiciando meios e estruturas adequadas (e mais convenientes) ao suporte de tais processos de análise, uma vez que são capazes de responder a todos os seus requisitos funcionais (e por vezes mesmo operacionais) inerentes à área de negócio e mercado no qual se posicionam. Um agente de decisão pode, assim, encontrar nesse tipo de ferramentas um suporte adequado para o ajudar a analisar grandes volumes de dados, com diferentes perspetivas e granularidade, de forma rápida e breve (Cuzzocrea et al., 2009).

Os *data warehouses* conciliam dados de diversas fontes, mas muitas vezes devido à grande dimensão dos dados, as sessões de exploração são muito demoradas. Ora, respostas rápidas e precisas estão na base do sucesso. Os *data warehouses* oferecem a melhor oportunidade para a análise de dados e os sistemas *OLAP* são o veículo para suportar essas análises (Paulraj, 2001). Implementam vários serviços que respondem de forma adequada às interrogações colocadas pelos agentes de decisão. Estes tempos de resposta são obtidos porque simplesmente estas ferramentas já possuem os dados, ou parte destes, previamente materializados. Estes dados materializados encontram-se organizados numa estrutura conhecida por hipercubo, que tem muitas vezes como finalidade alimentar ferramentas e serviços de *reporting* e apresentação.

Um hipercubo consiste num conjunto de dimensões e medidas. As medidas são os valores que estão sobre análise e as dimensões as coordenadas para essas mesmas medidas. Uma dimensão pode estar organizada em hierarquias que representam diferentes níveis de agregação (Niemi el al., 2001). Após a materialização do cubo de dados, muitos dos dados não são levados em conta por parte dos agentes de decisão durante as suas sessões de exploração. Uma forma de determinar quais as células do cubo que devem ser materializadas é através da monitorização destas mesmas consultas, com o objectivo de averiguar quais as células mais solicitadas e importantes para dar resposta às questões lançadas por estes utilizadores. Este método de reestruturação e optimização de

um cubo de dados necessita de uma constante supervisão das sessões de exploração, de forma a realizar uma seleção eficaz das células.

1.2. Motivação e Objectivos

A motivação para o desenvolvimento desta dissertação teve como base a ambição de construir um algoritmo que sugerisse uma vista multidimensional de dados para materialização. Esta vista, deverá materializar uma região de forte probabilidade de incidência das queries provenientes de consultas dos utilizadores. O algoritmo a desenvolver, bem como o método acolhido, pretende apresentar-se como uma alternativa viável aos já existentes no domínio, tendo como finalidade tirar melhor partido dos recursos disponíveis e fazer a otimização dos tempos de resposta de um cubo de dados. O método idealizado assenta num processo de monitorização intensiva das sessões de consulta dos agentes de decisão, de forma a extrair um perfil de utilização. Este perfil será um alicerce importante no algoritmo a produzir, que permitirá, também, concluir quais as vistas mais solicitadas por um ou mais utilizadores, e como tal, fazer uma correspondência entre os dados e as necessidades dos agentes. Esta correspondência deve ser o mais fiel possível, de forma a evitar que a vista sugerida possa conter dados que não são relevantes. Se isto acontecer, então ocorrerá, obviamente, um desperdício de recursos. Mas, por outro lado, caso a vista não contenha os dados relevantes para os utilizadores, então estes dados terão de ser calculados, o que implicará um tempo de resposta mais elevado.

Assim, neste trabalho pretendemos realizar um estudo e desenvolver um algoritmo para fazer a seleção de células de um hipercubo de acordo com o tipo de utilização que estas tenham tido ao longo dos vários processos de exploração a que estiveram submetidas. Depois, a partir dos resultados produzidos pelo algoritmo, definir-se-á um plano de materialização adequado com base nas células que, de facto, são relevantes para os processos de resposta às interrogações colocadas pelos utilizadores, deixando de parte as células "mortas", ou seja, não relevantes. De forma mais concreta, pretendemos:

- Com base no perfil exploração dos utilizadores, perceber como identificar as células mais relevantes.
- estudar os algoritmos já desenvolvidos, de forma a retirar princípios e ideias que sejam uma mais valia para o desenvolvimento do algoritmo pretendido.
- desenvolver um algoritmo que permita a identificação e seleção das células do cubo mais utilizadas, com base num conjunto de critérios de utilidade e de exploração.
- analisar e validar o algoritmo desenvolvido, comparando-o com outros algoritmos similares, bem como verificar a sua utilidade prática.

1.3. Organização do Documento

Além do presente capítulo, está dissertação integra mais quatro capítulos, nomeadamente:

- **Capítulo 2 - OLAP, Sistemas de Processamento Analítico**. Neste capítulo é feita uma breve abordagem aos sistemas de processamento analítico, desde os conceitos mais elementares até à sua arquitectura base.
- **Capítulo 3 – Esfoliação de Hipercubos**. Aqui é apresentado o método proposto para fazer a identificação das células de dados de um cubo que raramente são utilizadas e com bas nessa informação estabelecer uma estratégia para a sua "eliminação" e consequente redução dos recursos computacionais envolvidos no processamento e materialização de um cubo. Primeiramente é anunciado uma estensão às Classes de Equivalência e como estas podem auxiliar na indentificação dos dados mais relevantes para os utilizadores. De seguida é descrita a forma como as sessões OLAP são utilizadas para gerar Cadeias de Markov. No final, é feita uma conjugação destes dois elementos, de forma a criar o método a que nos propusemos .
- **Capítulo 4 – Aplicação desenvolvida e Análise de Resultados:** Neste capítulo é demonstrado a aplicação e o ambiente desenvolvidos com vista a demonstrar o algoritmo proposto anteriormente e apresentados alguns

testes que foram realizados sobre o algoritmo desenvolvido, assim como as análises efectuadas sobre os resultados obtidos.

- **Capítulo 5 – Conclusões e Trabalho Futuro**. Neste último capítulo, apresenta-se as conclusões sobre o trabalho efectuado e algumas linhas de orientação a seguir de forma a continuar a desenvolver o trabalho realizado no âmbito desta dissertação.

Capítulo 2

OLAP, Sistemas de Processamento Analítico

2.1. Sistemas de processamento analítico

A informação é cada vez mais um tópico de grande importância e interesse no mundo dos negócios, pois cada vez mais as organizações registam, armazenam e analisam maiores volumes de dados. Assim, armazenar e analisar dados relevantes tornou-se uma tarefa imperativa para o desempenho e para o planeamento do crescimento das empresas. Contudo, existem um conjunto de soluções direcionadas a responder a este tipo de desafios, sendo as duas mais comuns, os *data warehouse* e os cubos OLAP. Um *data warehouse* é simplesmente uma base de dados que contém informação que suporta a tomada de decisão e que é gerida de forma independente dos sistemas operacionais da empresa. Por outro lado, um cubo OLAP, é apenas uma outra palavra para um conjunto multidimensional de dados, sendo um mecanismo de interrogação de informação estruturada e organizada (Solver, 2015). Por vezes estes dois termos são usados de forma indiscriminada, porém são componentes diferentes de um sistema muitas vezes referenciado como um sistema de suporte à decisão.

Normalmente, um *data warehouse* e um sistema OLAP, nascem na mesma plataforma. Enquanto que estes dois têm um tamanho consideravelmente pequeno, é de interesse mantê-los juntos por questões de economia. Muito facilmente, dentro de um prazo de um ano, um *data warehouse* pode atingir um grande crescimento e quando isso acontece é necessário pensar em mover o sistema OLAP para uma plataforma independente, de forma a aliviar o congestionamento do outro sistema. Assim, para que se possa saber como e

quando mudar ou separar estes dois sistemas, existem as seguintes orientações (Paulraj Ponniah, 2001):

- Quando o tamanho e utilização do *data warehouse* escola e atinge o ponto onde este requer todos os recursos disponíveis da plataforma onde está alocado.
- Se muitos departamentos necessitam do sistema OLAP para análise de dados.
- Os utilizadores esperam que os sistema OLAP mantenha um determinado nível de desempenho. O *data warehouse* pode necessitar de ser carregado com mais dados e se este processo colocar em causa o desempenho do sistema OLAP, é necessário então recorrer à separação.
- Em organizações com informação descentralizada e com utilizadores em diferentes partes geográficas, é imperativa a existência de plataformas OLAP separadas.
- Se a ferramenta OLAP necessita de uma configuraçao diferente daquela que a plataforma do sistema *data warehouse*, então o sistema OLAP necessita de estar numa plataforma diferente, que esteja configurada de acordo com as suas necessidades.

Os sistemas de suporte à decisão incluem outras componentes, tais como bases de dados ou outras aplicações, que têm como finalidade providenciar meios para análise de informação de negócio (Microsoft TechNet, 2015). A figura 1 mostra a arquitetura básica de um sistema de suporte à decisão.

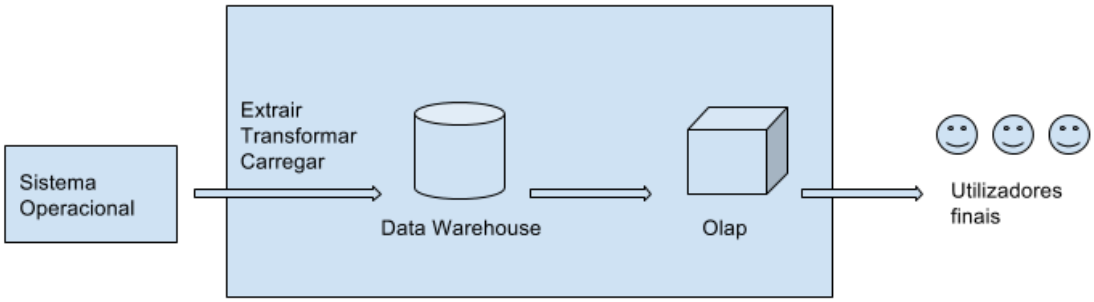


Figura 1 – Um sistema de suporte à decisão – figura adaptada de (Microsoft TechNet, 2015)

De forma mais simplificada, um *data warehouse* é uma base de dados que contém a informação que normalmente representa o histórico da atividade da organização. Estes dados são usados com o foco de tomada de decisão, isto é, com o propósito de serem consultados e não com o objetivo de suportar transações. A tecnologia OLAP providencia meios para que os dados existentes nos *data warehouses* sejam consultados de forma eficiente, assegurando respostas rápidas a queries complexas (Microsoft System Center, 2015). A importância desta, advém do facto do modelo multidimensional que suporta e das técnicas de organização de dados utilizadas – agregação e sumarização -, com o objetivo de proporcionar sistemas rápidos e flexíveis aos analistas (Microsoft TechNet, 2015). Em suma, estas aplicações de análise dimensional, tentam encontrar padrões anormais nos dados, sumarizando os valores das medidas de forma a extrair informação estatística e depois cruzar essa informação.

A contínua análise dos dados poderá ajudar no entendimento das mudanças que surgem no negócio das empresas, bem como auxiliar na identificação das soluções para os problemas que surgem na sua evolução e também no planeamento estratégico futuro das organizações (Gray el al., 1995). O modelo multidimensional foi introduzido por Codd em 1970, que lhe valeu um prémio uma década mais tarde, servindo de base a uma indústria de base de dados multibilionária. Nesta última década o modelo multidimensional emergiu ainda mais, quando a nossa preocupação passou para a análise de dados, "relaxando-se" a atenção no processar transações relacionadas com atividades de funcionamento das empresas (Pedersen et al., 2001).

As bases de dados multidimensionais olham para os dados como cubos de várias dimensões que suportam hierarquias nos atributos das suas dimensões. Um cubo é uma estrutura de dados que contém dois tipos base, as medidas, que são dados numéricos, como quantidades e médias que auxiliam na tomada de decisão, e as dimensões, que podem ser vistas como categorias que tem como objetivo organizar as medidas (Microsoft, 2015). As hierarquias tem como propósito identificar um caminho de navegação para uma dimensão, de tal modo que os valores das medidas nos diferentes níveis de agregação, sendo estes níveis chamados de hierarquias, conseguem ser obtidos através de operações de *roll-up* e *drill-down* (Anna Rozeva, 2007). As hierarquias não são mais nem menos do que caminhos lógicos de atributos de uma dimensão, em que cada nível desse caminho proporciona vistas diferentes para a análise dos dados.

2.2. Implementações OLAP

Existem quatro principais abordagens para implementar sistemas OLAP. Estes sistemas estão classificados segundo a arquitetura usada para guardar e processar os dados multidimensionais. Vejamos então cada um dessas abordagens.

ROLAP – Relational OLAP

Os sistemas ROLAP usam a tecnologia das base de dados relacionais para armazenar os dados e construir estruturas especializadas de índices com o objetivo de conseguir grandes ganhos de performance. A abordagem ROLAP tira partido da criação de uma camada de metadados que permite a criação de vistas sobre os dados multidimensionais e que tem como vantagem a não necessidade de ter um modelo multidimensional estático. A arquitetura de um sistema destes e as desvantagens encontradas neste tipo de abordagem estão apresentadas na figura 2 e têm as seguintes características:

- Problemas de desempenho relacionados com o processamento de queries complexas, as quais requerem o tratamento de dados da base de dados relacional ao longo de várias etapas.

- Desenvolvimento de uma camada intermediária, de forma a facilitar o desenvolvimento da própria aplicação multidimensional.

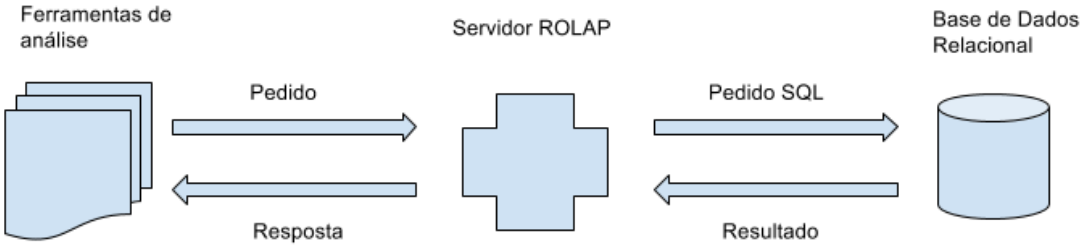


Figura 2 - Arquitetura de um sistema ROLAP – figura adaptada de (Connolly & Begg, 2005).

A ideia de adotar tecnologia relacional para este fim é bastante plausível se for tomada em atenção a quantidade de literatura que existe sobre base de dados relacionais. Contudo esta tecnologia não abrange os conceitos de dimensão, medida ou hierarquia, sendo necessário o uso de várias técnias de forma a representar o modelo multidimensional (Golfarelli & Rizzi, 2009).

MOLAP – Multi-Dimensional OLAP

Os sistemas MOLAP guardam os dados em disco fazendo uso de estruturas multidimensionais especializadas. Estes sistemas muitas vezes incluem mecanismos para lidar com cubos esparsos e aplicam técnicas de indexação, de forma a localizar rapidamente os dados aquando do lançamento de queries. Também por questões de desempenho na resposta a queries, os dados são tipicamente guardados e armazenados segundo previsões de utilização. Estes sistemas desempenham operações de forma muito mais eficiente e natural que as implementações ROLAP. A arquitetura de um sistema destes e as desvantagens encontradas neste tipo de abordagem estão apresentadas na figura 3 , e apresenta as seguintes características:

- Uma vez que os dados estão guardados segundo certos critérios previamente definidos, a sua análise pode estar comprometida no sentido

em que eles poderão ser novamente reestruturados, de forma a refletir novas necessidades ou requisitos de análise.

- Apenas parte dos dados podem ser guardados e analisados de forma eficiente, visto que as estruturas que os suportam são limitadas no acesso a dados detalhados.

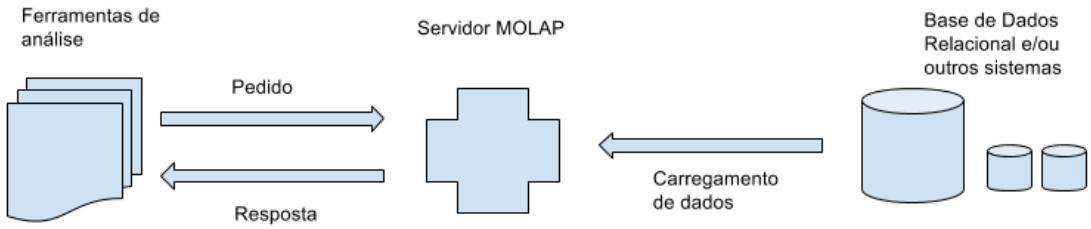


Figura 3 - Arquitetura de um sistema MOLAP – imagem adaptada de (Connolly & Begg, 2005).

HOLAP – Hybrid OLAP

Os sistemas HOLAP surgem da combinação dos sistemas ROLAP e MOLAP, tentando trazer o melhor dos dois, conjugando grandes capacidades de armazenamento com baixos tempos de resposta. Estas ferramentas fornecem dados tanto a partir de uma base de dados relacional como através de servidores MOLAP para um *desktop,* no qual são guardados e mantidos localmente para análise. Os sistemas MOLAP são caracterizadas por serem ferramentas simples de instalar, de custos reduzidos. A arquitetura de um sistema destes e as desvantagens encontradas neste tipo de abordagem estão apresentados na figura 4 (Connolly & Begg, 2005) e têm como características o seguinte:

- Apenas uma parte dos dados consegue ser eficientemente gerida e mantida.
- A possibilidade dos utilizadores manterem um cubo específico, à sua medida, faz com que surjam problemas de inconsistência entre os dados dos vários utilizadores.

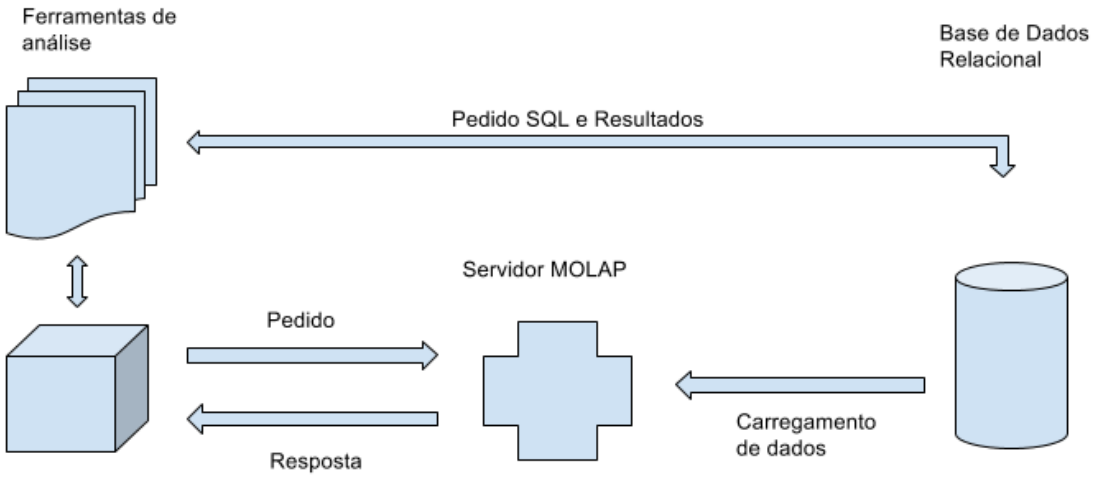


Figura 4 - Arquitetura de um sistema HOLAP– imagem adaptada de (Connolly & Begg, 2005).

DOLAP – Desktop OLAP

Os sistemas DOLAP têm vindo a crescer nos últimos tempos. Estas ferramentas fazem o processamento dos dados OLAP usando uma aplicação multidimensional nas próprias plataformas do cliente, de forma a fazerem parte ou grande parte dos cálculos necessários para análise da informação. A administração das bases de dados DOLAP são feitas por um servidor central que prepara os cubos de cada cliente, permitindo a estes o acesso e a leitura à informação. A arquitetura de um sistema destes e as desvantagens encontradas neste tipo de abordagem estão apresentadas na figura 5 (Connolly & Begg, 2005), sabendo-se que:

- É necessário haver algum tipo de poder de processamento do lado dos clientes.
- É imperativo tomar medidas de segurança que controlem o sistema DOLAP. Uma vez que os dados são extraídos do sistema pelos utilizadores, o mecanismo de segurança é geralmente implementado de forma a limitar a quantidade de informação de cada cubo.

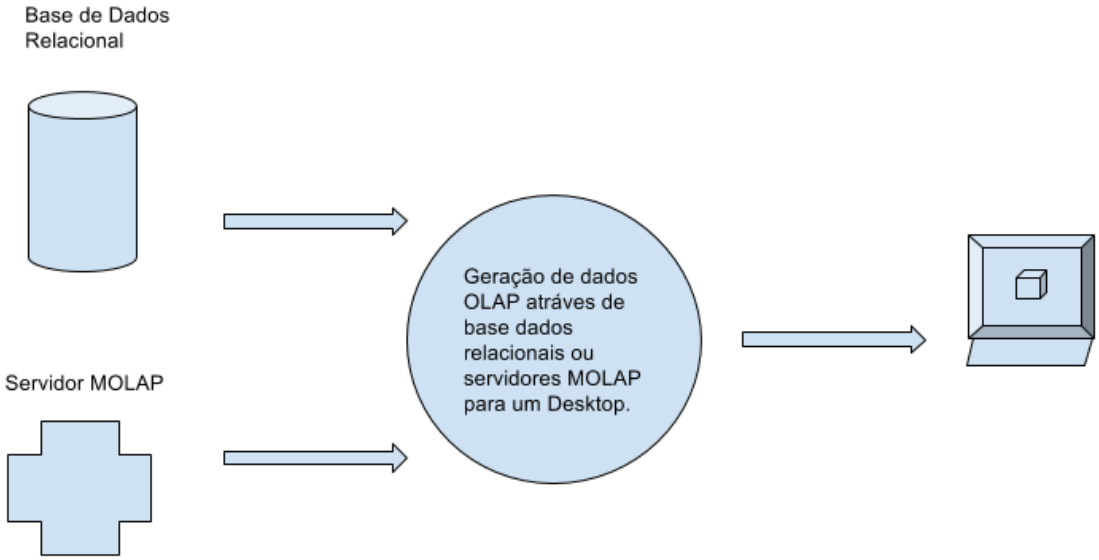


Figura 5 - Arquitetura de um sistema DOLAP– imagem adaptada de (Connolly & Begg, 2005).

Edgar F. Codd escreveu um artigo em 1985 no qual definia as regras para os sistemas relacionais de gestão de base de dados. Este artigo revolucionou a indústria de tecnologias da altura. Mais tarde, em 1993, Codd trabalhou de forma a definir as regras para as ferramentas OLAP (Olap, 2015) .Algumas das regras mais importantes definidas então para estas ferramentas foram:

- **Vistas multidimensionais** – As ferramentas devem ser capazes de apresentar vistas multidimensionais do modelo que corresponde à visão que os utilizadores tem das organizações, bem como devem ser intuitivas e fáceis de utilizar.
- **Transparência –** Devem ser providenciados ambientes de apresentação de informação familiares aos utilizadores, ou seja, todas as tecnologias relacionadas com servidores, arquiteturas e fontes de alimentação destas ferramentas devem ser transparentes aos utilizadores.
- **Acessibilidade –** Estas ferramentas devem ser capaz de capturar dados de diversas fontes de dados.

- **Desempenho consistente –** À medida que aumenta o número de dimensões, níveis agregação e dados, os utilizadores não devem ser capazes de detetar mudanças significativas no desempenho do seu

sistema. O sistema deve ser suficientemente robusto para suportar estas mudança no sistema.

- **Arquitetura de cliente –** Um sistema OLAP deve ser capaz de funcionar de modo eficiente num ambiente proporcionado pelo sistema do cliente. A arquitetura deve ser capaz de proporcionar desempenho, flexibilidade, capacidade de adaptação, escalabilidade e interoperabilidade.
- **Suporte para vários utilizadores –** Um sistema OLAP deve ser capaz de suportar vários utilizadores que trabalhem de forma simultânea e específica sobre um mesmo conjunto de dados.

2.3. Sessões e Operadores OLAP

Basicamente, uma sessão OLAP pode ser definida como um caminho de navegação que corresponde a um processo de análise de factos, de acordo com diferentes pontos de vista e diferentes detalhes de informação. Este caminho não é mais que uma sequencia de queries, na qual cada query está associada de alguma forma à query anterior. Cada passo de uma sessão de análise é caracterizado por um operador OLAP, que transforma a query anterior numa outra query nova (Golfarelli & Rizzi, 2009). Os operadores OLAP mais comuns são: *roll-up, drill-down, slice and dice*, *pivot, drill-across* e *drill-throught*.

Roll-Up e Drill-Down

O operador *roll-up* causa um incremento na agregação de dados e remove um nível de detalhe da hierarquia. Isto é, permite navegar para cima nos níveis de hierarquia para uma dada dimensão, o que resulta um posicionamento numa zona de menor detalhe (Athena, 2015) .A figura 6 mostra o exemplo de hierarquias em atributos de dimensões (Golfarelli & Rizzi, 2009).

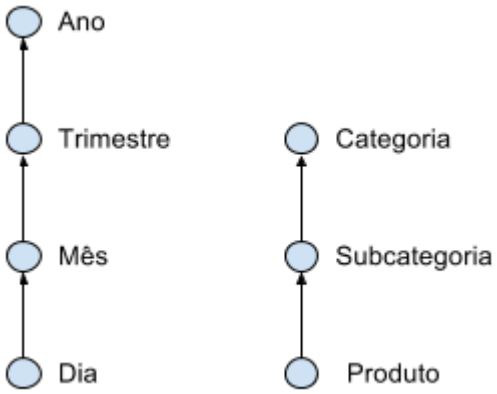


Figura 6 - Exemplo de hierarquias em atributos de dimensões

Esta operação, tal como a operação de *drill-down,* consiste em mudar as agregações dos valores das medidas, que no caso da *roll-up* consiste em agregá-los *e que* no caso da *drill-down* consiste em detalhar e desagregar esses mesmo valores*.* Estes dois operadores são complementares (D. Boukraa et al.).

Slice and Dice

A operação de *slice* produz um cubo OLAP fatiado, permitindo ao analista escolher um determinado valor para um dada dimensão. Por exemplo, para uma dimensão de Calendário, igualar o ano a "2012", permite que esta dimensão seja removida e todas as restantes dimensões apresentem apenas os dados para essa mesma data. A operação de *dicing* produz um subcubo, no qual o analista escolhe múltiplos valores para múltiplas dimensões, por exemplo, igualando o ano a "2012" e "2013", assim como a localização a "Eindhoven" e "Amsterdam". Nenhuma dimensão é removida, apenas as métricas desse dado cubo para esses filtros são consideradas (Wil et al.). Na figura 7, podemosobservar uma ilustração de um exemplo de uma operação de *slicing* (em cima) e de uma operação de *dicing* (em baixo) sobre um cubo de dados*.*

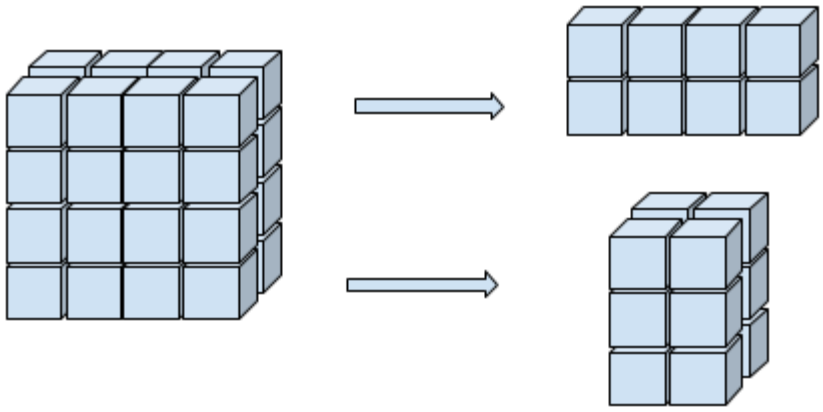


Figura 7 – Ilustração da operação de slice and dice – figura adaptada de (Golfarelli & Rizzi, 2009).

Pivot

Também conhecida como *rotation,* como o nome indica, esta operação implica uma rotação e consequentemente uma mudança na orientação dos eixos dimensionais do cubo (Athena, 2015) .Na prática, esta operação permite realizar diferentes combinações de dimensões tal como mostra a figura 8.

Category/Revenue		North	South
Books	**2014**	416.183$	137.502$
	2015	534.932$	138.683$

Category/Revenue	North		South	
	2014	**2015**	**2014**	**2015**
Books	416.183$	534.932$	137.502$	138.683$

Figura 8 – Iustração da utilização da operação Pivot de forma a obter diferentes combinações de dimensões – figura adaptada de (Golfarelli & Rizzi, 2009)

Drill-Across e Drill-Throught

O termo *drill-across* tem como significado a oportunidade de criar uma ligação entre dois ou mais cubos relacionados com o objetivo de cruzar e comparar dados. Dois cubos com dados relativos a vendas e descontos, por exemplo,

podem ser usados para cruzar informação e perceber uma potencial relação entre os dois (Golfarelli & Rizzi, 2009). A operação *drill-throught* faz a ligação entre os dados agregados nos *data marts* com os dados presentes nos sistemas operacionais ou outras camadas intermédias.

2.4. Benefícios dos Sistemas OLAP

O principal benefício das aplicações OLAP é a consistência da informação e da forma como esta é processada. Não importa qual é a dimensão ou o volume dos dados, os servidores OLAP deverão sempre apresentar a informação de forma consistente, de tal modo que os analistas saberão sempre onde e como a encontrar. Isto torna-se especialmente relevante para comparar dados antigos com dados mais recentes (Olap House, 2015) .Os negócios das empresas prosperam ou caem de acordo com a sofisticação e rapidez dos seus sistemas de informação e na habilidade de analisar ou sintetizar a informação que possuem nesses sistemas, uma vez que o número de indivíduos nas organizações que tem necessidade de explorar dados está em permanente crescimento (Codd el al., 1993). As vantagens que se poderão atingir com a adoção e exploração de um sistema OLAP, bem implementado, são (Golfarelli & Rizzi, 2009), (Connolly & Begg, 2005):

- O aumento da produtividade e poder de decisão dos agentes de análise, visto que toda a informação se encontra facilmente acessível e num formato mais familiar, tornando mais eficientes os processos de decisão da organização.
- A redução de tarefas de produção de relatórios por parte das aplicações de desenvolvimento das organizações, visto que agora os analistas possuem ferramentas próprias para gerar e modelar tais relatórios.
- O aumento das receitas e diminuição dos custos da organização, uma vez que a ferramenta OLAP permitirá uma melhor e mais rápida resposta às mudanças que se dão nos mercados.
- A diminuição da dependência da equipa de desenvolvimento por parte dos agentes de decisão, pois que, como dito anteriormente, as

ferramentas OLAP proporcionam ambientes de exploração intuitivos e familiares aos seus utilizadores.

- O aumento da confiança da informação, visto que frequentemente estes sistemas têm como fontes de dados um sistema de *data warehousing.*
- A diminuição da carga de outros sistemas devido ao carregamento de grande parte dos dados para os servidores OLAP.
- A retenção do control organizacional sobre a integridade dos dados, uma vez que as aplicações OLAP estão dependentes dos dados que estão armazenados num *data warehouse* e cujos os sistemas operacionais são responsáveis por os alimentar.
- O aumento do retorno e lucro das organizações, pois estes sistemas permitem que os processos de tomada de decisão se desencadeie de forma mais rápida e eficaz.

Capítulo 3

Esfoliação de Hipercubos

3.1 Classes de Equivalência

As classes de equivalência são um conceito proveniente da matemática e tem dado aso a outros tipos de estudos e aplicações. Niemi, Nummenmaa e Thanisch em 2001, propuseram o uso desta técnica para encontrar similaridades entre queries MDX, com a finalidade de propor um cubo novo que refletisse melhor as necessidades de exploração. De acordo com esses autores, duas queries são similares se partilham uma ou mais dimensões de forma direta ou transitiva (Niemi el al., 2001). Desta forma é possível distribuir as queries por conjuntos, em cada um desses conjuntos contém as queries que atuam sobre um determinado domínio do hipercubo. A estes conjuntos dá-se o nome de classes de equivalência. Este método permite identificar diferentes grupos de queries, nos quais essas mesmas queries tem entre si interseções nos resultados. Por outras palavras, o este método permite realizar é agrupar as queries que atuam sobre uma determinada área do cubo de dados. A figura 9 ilustra a forma como o método identifica as classes de equivalência. Pela aplicação deste método as classes de equivalência a identificar nessa figura seriam as:

- E1 = {Q1, Q2, Q3}
- E2 = {Q4, Q5}
- E3 = {Q6, Q7, Q8}

Isto porque, como é possível verificar, elas intersectam-se na medida em que o conjunto de resultados de cada query intersecta-se com uma ou mais queries

desse grupo. Ou seja, como as classe de equivalência se intersectam direta ou transitivamente entre si.

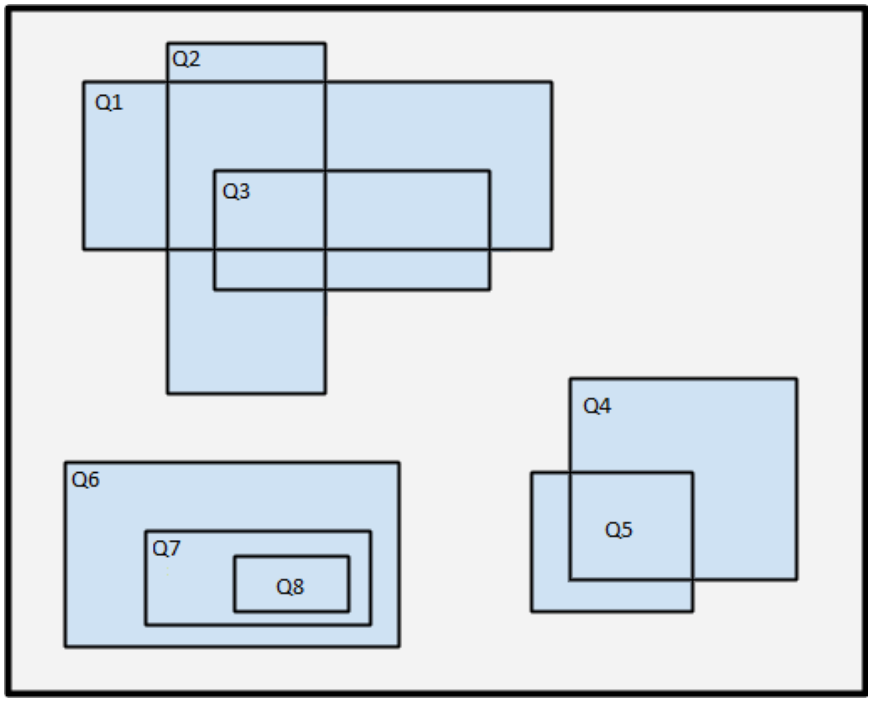


Figura 9 - Identificação das classes de equivalência – figura extraída de (Niemi et al., 2001)

De forma a apresentar uma extensão à técnica apresentada em 2001 por Niemi et al. (2001), apresentamos primeiramente uma descrição exemplificada do algoritmo proposto por estes autores. Considere-se, então, um conjunto Q de queries MDX que foram lançadas sobre um determinado hipercubo. As queries consideradas nesse conjunto foram as seguintes:

Q1 - Select product_id Where quantity, profit;

Q2 - Select product_group Where profit, year.[2000];

Q3 - Select year, product.[Food] Where quantity;

Q4 - Select employee_id Where quantity, profit;

Q5 - Select employee_id, customer_id Where quantity, profit;

Q6 - Select store.[USA] Where quantity, profit;

Q7 - Select employee_id, promotion Where profit;

Q8 - Select promotion.[USA] Where profit, quantity;

O algoritmo de Niemi et al. (2001) desenvolve-se em quatro etapas distintas, nomeadamente:

1) Para cada query Qi em Q, colocar os atributos da cláusula Select no conjunto Xi. Ignorar dimensões demasiado gerais como tempo ou geografia. Desta forma obtemos os seguintes conjuntos:
    a. X1 = {product id}
    b. X2 = {product group}
    c. X3 = {product}
    d. X4 = {employee id}
    e. X5 = {employee id, costumer id}
    f. X6 = {store}
    g. X7 = {employee id, promotion}
    h. X8 = {promotion}.

2) De seguida, com base nas dependências funcionais de cada um dos conjuntos Xi estabelecidos, deve-se construir o conjunto Yi que conterá a chave da dimensão K, se e só se existir um atributo A em Xi, tal que A pertença aos atributos de K. Após a realização deste passo ficamos com os conjuntos
    a. Y1 = {product id}
    b. Y2 = {product id}
    c. Y3 = {product id}
    d. Y4 = {employee id}
    e. Y5 = {employee id, costumer id}
    f. Y6 = {store id}
    g. Y7 = {employee id, promotion id}
    h. Y8 = {promotion id}

3) **Num terceiro passo, faz-se a** construção das classes de equivalência da seguinte forma: duas queries Q e Q' pertencem à mesma classe de equivalência E se for possível formar uma sequência de queries <Q0 = Q, Q1,

..., Qn, Q' = Qn+1> tal que Yi ⋂ Yi+1 ≠ ∅ e 0 ≤ i ≤ n, em que Yi representa o conjunto de chaves de dimensão de Qi. O resultado produzido neste terceiro passo é constituído pelas classes de equivalência:

a. E1 = {Q1, Q2, Q3}
b. E2 = {Q4, Q5, Q7, Q8}
c. E3 = {Q6}

Além destes três passos, podemos realizar um outro passo adicional, que consiste em construír uma nova classe de equivalência E' = {Q'}, para cada classe de equivalência Ei, para cada Q em Ei, se existir uma query Qi em Ei, com as mesmas dimensões que Q, mas num nível de detalhe diferente de Q.

3.2 Extensão das Classes de Equivalência

Como visto anteriormente, o algoritmo proposto por Niemi et al. (2001) permite dividir as queries por diferentes grupos, em que em cada grupo as queries atuam sobre um determinado domínio do hipercubo. Contudo, nestes domínios é possível identificar as zonas mais afectadas, isto é, os dados mais relevantes para responder às queries que pertencem a cada classe. Ou seja, é possível identificar quais as zonas dentro de uma determinada classe de equivalência que têm maior incidência por parte das queries dessa mesma classe. Com isto é possível perceber quais os dados mais relevantes que essa classe identifica. A figura 10 exemplifica essa idea, mostrando diferentes gradientes de cores, mais pesados, para áreas nas quais as queries têm maior atuação e zonas mais claras nas quais a atuação das queries dessa classe de equivalência têm menor impacto.

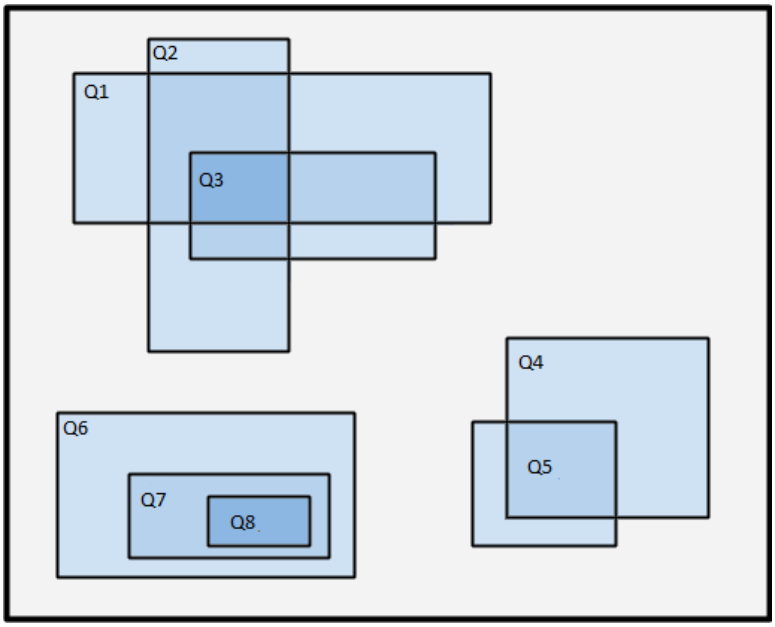


Figura 10 – Ilustração da extensão às classes de equivalência com identificação de sobreposição de domínios.

Como forma de responder a esta questão concebemos uma extensão ao método das classes de equivalência, de forma a identificar as zonas mais relevantes dentro de cada classe de equivalência. A forma como este processo de extensão atua pode ser descrito da seguinte maneira. Para cada classe de equivalência Ei obtida deve-se obter o conjunto Zi das chaves de dimensões, dado pela união de todos os conjuntos Yi das queries em Ei. Depois, para cada chave de dimensão em Zi, obter o par (chave de dimensão, Ti), onde Ti é o número de vezes que uma dada chave de dimensão ocorre em Yi. Ao se aplicar este passo, as classes de equivalência mantêm-se em relação ao que aconteceu no passo anterior, mas, agora, é possível perceber quais as dimensões mais relevantes dentro das classes de equivalência estabelecidas. No formato estendido as classes de equivalência são:

a. E1 = {(product id, 3)}
b. E2 = {(employee id, 3), (costumer id, 1), (promotion id, 2)}
c. E3 = {(store id, 1)}.

Numa perspetiva um pouco mais simplificada, o que acontece é que o número de vezes que uma dada dimensão serviu para relacionar as queries de uma classe de equivalência é registado. Atentamos na classe de equivalência E2, o que se

verifica é que a chave de dimensão employee id caracterizou o relacionamento entre as queries Q4, Q5 e Q7, o que significa que é muito relevante na identificação dessa mesma classe de equivalência.

3.3 Sessões OLAP e Cadeias de Markov

Um utilizador OLAP segue muitas vezes uma dada linha de análise e de prospecção de dados, na qual faz um conjunto de queries usualmente relacionadas entre si sobre um dado hipercubo. Qualquer utilizador OLAP deve ser capaz e ter os meios necessários para fazer a costumização de um hipercubo, de forma a este estar mais ajustado e orientado para as suas necessidades mundanas de tomada de decisão (Niemi et al., 2001). Isto é especialmente importante se tivermos em conta que os dados do data warehouse que atua como fonte podem estar em constante atualização e, com tal, é necessário que o cubo reflita tais mudanças através de um refrescamento mais frequente.

Dado um conjunto de queries de um determinado utilizador, lançadas ao longo das suas sessões de exploração de dados, é possível estudar e traçar o comportamento deste mesmo utilizador. As cadeias de Markov endereçam bem este tipo de problema, pois permitem analisar processos estocásticos e visualizá-los sob a forma de um grafo (Deshpande et al., 2004). As queries realizadas por um dado utilizador podem ser então utilizadas para alimentar uma cadeia de Markov e assim tornar possível a identificação do comportamento e da forma de raciocínio do agente de decisão à qual a cadeia corresponde. As cadeias de Markov apresentam a característica de serem incrementais. Isto significa que é possível acrescentar nova informação à cadeia, sem esta ter de ser recalculada (Borges et al., 1999) (Sarukkai, 2000).

Considere-se, por exemplo, C como um conjunto de queries lançadas por um determinado utilizador U, em que cada query representa um determinado estado da cadeia, de forma a que C = {Qi, ..., Qn} e $0 < i < 9 = n$. No quadro apresentado na tabela 1 podemos ver um exemplo de um conjunto de sessões relativas a um dado utilizador U, bem como o conjunto das queries lançadas durante a sessão.

Sessão	Estrutura da sessão
1	Q1->Q3->Q2
2	Q1->Q2->Q4->Q5
3	Q4->Q6->Q7
4	Q6->Q8->Q7->Q2
5	Q7->Q6->Q8->Q4

Tabela 1 - Sessões de exploração e respetivas sequências de queries.

A probabilidade de um estado a' preceder o estado b', P(a',b') é dado pela razão entre o número de vezes em que a' antecede b' - |a', b'| - e o número de ocorrências de a' - |a'|. Ou seja, P(a',b') = |a', b'| / |a'|. Pressupondo o estado inicial como sendo o estado 0 e o estado final como sendo o estado -1, a probabilidade de partir do estado inicial 0 para outro estado, digamos a', é determinado pela razão entre o número de vezes que a' é a primeira query a ser lançada numa sessão e o número total de sessões, ou seja, P(0,a') = |a'| / (número total de sessões). O grafo da cadeia de Markov gerado a partir do exemplo apresentado na tabela 1 está apresentado de seguida na figura 11.

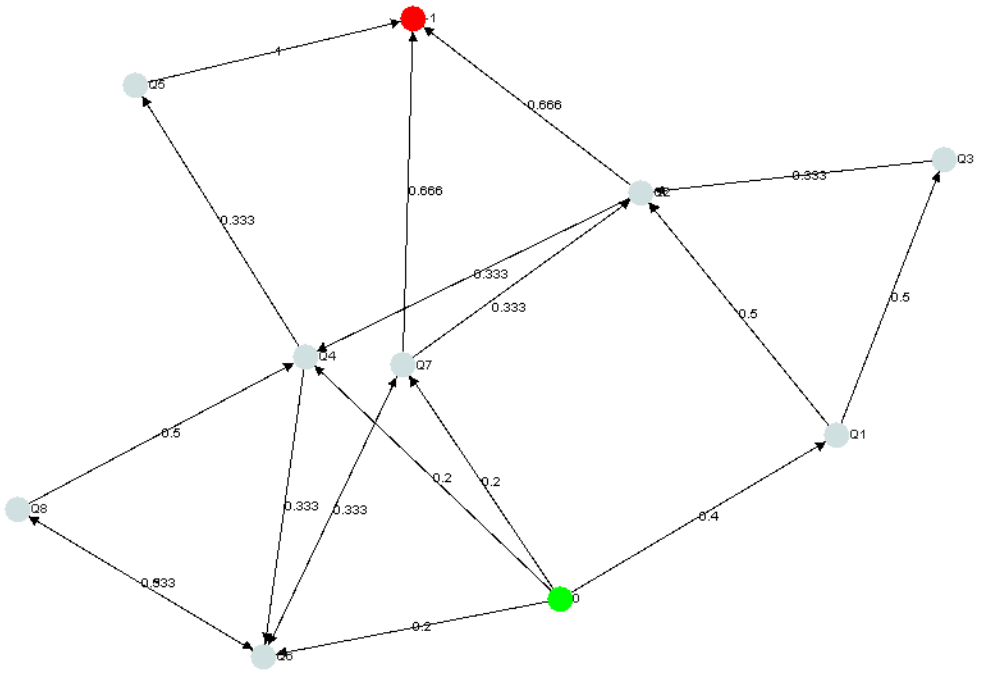


Figura 11 – Exemplo de uma cadeia de Markov.

Como é possível observar, o grafo apresentado na figura 11 inclui as probabilidades de passagem de estados, no qual cada query é identificada por um nodo e os estados de início e fim são representados por 0 e -1, respetivamente. Com isso é possível calcular o percurso cujos arcos entre estados esteja acima de um dado *threshold* e determinar o padrão mais frequente de pesquisa de um ou vários utilizadores. Por outras palavras, dessa forma é possível descobrir quais são as queries mais importantes nos processos de exploração OLAP e, como consequência, descobrir quais os dados mais relevantes para os utilizadores em questão.

3.4 Método para Esfoliação de Hipercubos

O algortimo proposto nesta dissertação assenta na conjugação dos dois métodos apresentados anteriormente. Assim, as cadeias de Markov para gerar grafos de precedência de queries baseadas em probabilidades, apresenta-se, simplesmente, como primeiro passo a ser aplicado. De seguida, identifica-se um caminho de maior probabilidade, bem como o conjunto de queries que fazem parte do grafo que foi obtido dessa forma. A próxima iteração do algortimo, passa por aplicar sobre esse conjunto de queries o método de extensão das classes de equivalência e dessa forma obter uma query que represente uma vista a materializar do cubo esfoliado. Todos estes passos serão explicados de seguida.

Os dados utilizados na descrição do algoritmo de extensão são os mesmos que foram usados na exemplificação dos métodos anteriores expostos neste mesmo capítulo. Assim sendo, prestemos atenção, novamente, na cadeira de Markov gerada anteriormente.

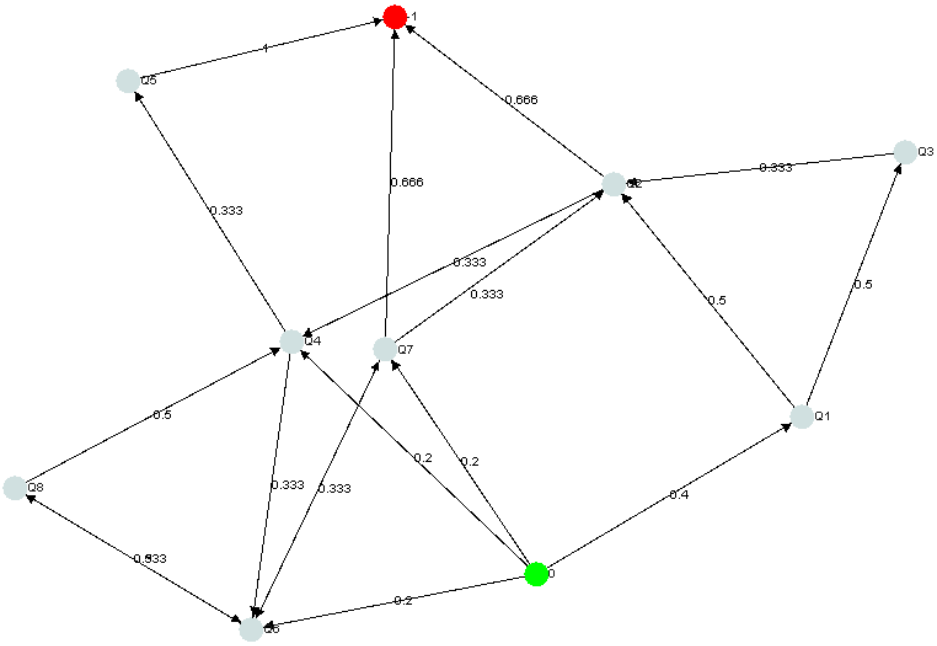


Figura 12 – A cadeia de Markov produzida

Ora, como referido anteriormente, para um conjunto de sessões, a geração da cadeia de Markov representa o primeiro passo no algoritmo proposto. O segundo passo, passa por definir um *threshold* para limitar a cadeia a um conjunto mais restrito de nodos, que representam grande parte das queries mais frequentes e correspondentemente aos dados mais solicitados pelos utilizadores, uma vez que o resultado de uma query são dados. tendo como base de que o valor do *threshold* a aplicar é de 40%, a cadeia resumir-se-á então ao que a figura 13 apresenta. Estando identificada a sub-cadeia que respeita o *threshold*, obtemos então o conjunto de queries C = {Q1, Q2, Q3, Q5, Q7}.

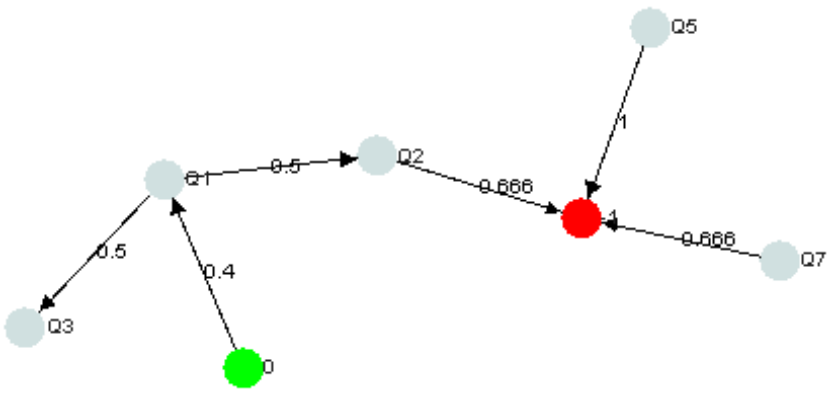


Figura 13 – A cadeia de Markov produzida para um threshold de 40%

Agora é a altura de aplicar o método de extensão às classes de equivalência sobre o conjunto C. De acordo com o apresentado anteriormente, na secção 3.1, no passo 1, obtemos os seguintes conjuntos:

- X1 = {product id}
- X2 = {product group}
- X3 = {product id}
- X5 = {employee id, costumer id}
- X7 = {employee id, promotion}

Depois, aplicando o passo 2, ficamos com os conjuntos:

- Y1 = {product id}
- Y2 = {product id}
- Y3 = {product id}
- Y5 = {employee id, costumer id}
- Y7 = {employee id, promotion id}

Seguidamente, aplicando o passo 3, as classes de equivalência obtidas são, então, as seguintes:

- E1 = {Q1, Q2, Q3}
- E2 = {Q5, Q7}

Por fim, no passo 4, o passo final, são obtidos as classes no formato estendido E1 = {(product id, 3)} e E2 = {(employee id, 2), (costumer id, 1), (promotion id, 1)}, aplicando a extensão ao método das classes de equivalência,.

O último passo do algoritmo de esfoliação tem como objetivo a sugestão de uma query que represente uma vista a materializar do hipercubo de trabalho, sendo essa vista o cubo esfoliado.

Tendo agora as classes de equivalência estendidas é então necessário escolher qual o fator de incidência dos atributos de dimensão a serem utilizados, de forma a gerar a almejada query. O valor deste fator será escolhido pelos utilizadores.

Admitindo, um valor para esse fator de 2, então a partir das classes obtidas sabemos que os atributos de dimensão que respeitarão tal valor são, nomeadamente, o "product id" e o "employee id". Assim sendo, a query seguiria uma estrutura como a seguinte:

```
SELECT product id, employee id FROM hypercube;
```

Contudo, este método também permite fazer a leitura dos filtros das queries obtidas na rede. Por exemplo, a query Q3 apresenta o filtro product.[Food], que permite a identificação da dimensão em causa, bem como o filtro e um fator de incidência. Neste caso, em particular, obteriamos o tuplo (Product, Food, 1). Se outras queries apresentassem filtros iguais, então seria feita uma junção entre si de modo a dar uma visão mais geral. Se ocorresse uma intersecção de filtros, o fator de incidência seria incrementado. Assim, para o exemplo apresentado, se o fator de incidência assumisse o valor 1, a query formulada passaria então a ser algo do género:

```
SELECT product id, employee id FROM hypercube WHERE product.[Food];
```

Este algoritmo engloba também as métricas das queries MDX lançadas pelos utilizadores. Porém, por uma questão de simplificação não serão apresentadas nesta descrição. No capítulo seguinte é apresentado um programa desenvolvido que cria as classes de equivalência estendidas para as métricas das queries e escolhe um fator de incidência para utilizar na geração da query da vista a materializar.

Capítulo 4

Sistema Piloto e Análise de Resultados

4.1 Tecnologias Utilizadas

Como já referido, o algoritmo proposto requer como input os dados relativos às diversas sessões realizadas pelos utilizadores, bem como as queries MDX lançadas durante esses processos. Sendo assim, a aplicação desenvolvida parte do pressuposto da existência de dois data marts, um que contém a informação relativa às sessões realizadas e um outro com informação relativa às queries MDX lançadas pelos utilizadores sobre um determinado hipercubo. Os dois data marts que suportam a aplicação desenvolvida encontram-se assentes num sistema operacional MySql, que foi escolhido por ser um sistema open source e ser facilmente integrável com outras tecnologias existentes no seu domínio de trabalho. A aplicação foi desenvolvida na linguagem de programação Java, uma linguagem madura e bem divulgada, com uma vasta API pública e que conta com várias contribuições de grandes empresas do sector das tecnologias, como a Google ou a Sun.

4.2 Data Marts

Comecemos a nossa descrição pelo data mart que suporta a informação relativa às sessões OLAP praticadas pelos utilizadores. Este data mart disponibiliza os dados necessário para realizar a análise temporal das sessões para um determinado utilizador e para um dado período do dia. Neste data mart estão integradas as seguintes dimensões: "Utilizador" ("dim_user"), "Período" ("dim_period") e "Calendário" ("dim_date"). Quanto à sua tabela de factos

("tf_session"), esta tem registada a informação sobre a hora de começo e fim da sessão, bem como o utilizador, o período associado e a data da sessão. Com este data mart é possível, por exemplo, aplicar o algoritmo de esfoliação de hipercubos para um dado dia da semana ou para um determinado período do dia, sobre as sessões realizadas por um ou mais utilizadores. Na figura seguinte apresenta-se o diagrama lógico do data mart "Sessões".

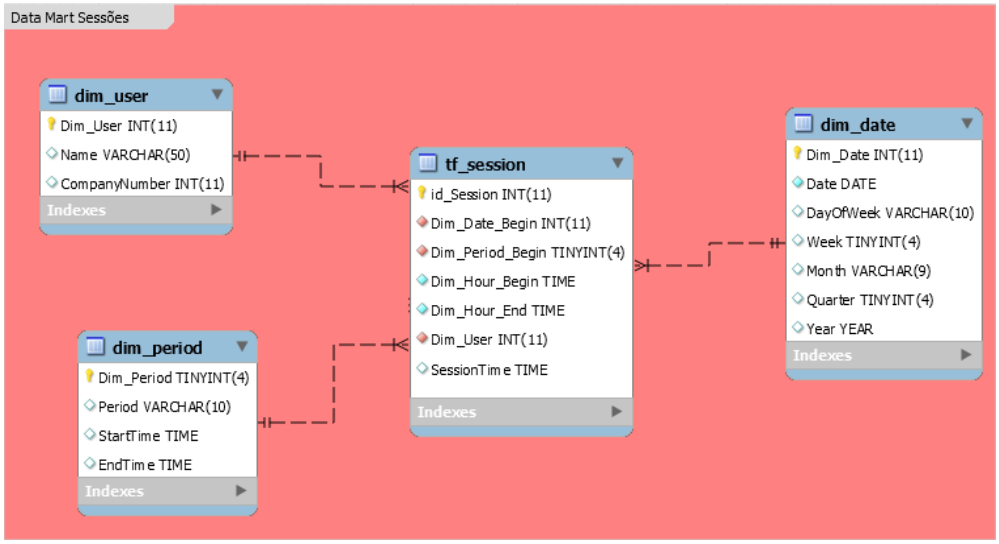


Figura 14 - Modelo lógico do data mart "Sessões".

A descrição pormenorizada do data mart "Sessões" relativa às suas tabelas de dimensão e tabela de factos pode ser encontrada de seguida nas tabelas2, 3, 4 e 5.

Tabela de factos de sessões: contém informação relativa às sessões praticadas pelos utilizadores durante as respetivas sessões de exploração OLAP.		
Atributo	**Descrição/Tipo**	**Exemplo**
Id_Session (PK)	Identificador da sessão. INT	1
Dim_Date_Begin (FK)	Data de início da sessão. INT	1
Dim_Period_Begin (FK)	Período de início da sessão. INT	1
Dim_Hour_Begin	Hora de início da sessão. TIME	8:30:00
Dim_Hour_End	Hora de fim da sessão. TIME	11:00:00
SessionTime	Tempo de duração da sessão. TIME	2:30

Tabela 2 - Descrição tabela de factos de sessões – "tf_session".

Dimensão utilizador: Contém informação relativa aos utilizadores que efetuam as sessões de exploração OLAP.		
Atributo	**Descrição/Tipo**	**Exemplo**
Dim_User (PK)	Identificador do utilizador. INT	1
Name	Nome do utilizador. Varchar	José Mourinho
CompanyNumber	Identificador da empresa do utilizador. INT	1

Tabela 3 - Descrição da dimensão utilizador – "dim_user".

Dimensão período: Contém informação relativa aos períodos dos dia.		
Atributo	**Descrição/Tipo**	**Exemplo**
Dim_Period (PK)	Identificador do período. INT	1
Period	Nome do período. VARCHAR	Afternoon
StartTime	Hora de início do período. TIME	12:00:00
EndTime	Hora de fim do período. TIME	19:59:59

Tabela 4 - Descrição da dimensão período – "dim_period".

Dimensão calendário: Contém informação relativa ao calendário.		
Atributo	**Descrição/Tipo**	**Exemplo**
Dim_Date (PK)	Identificador da data. INT	1
Date	Data. DATE	2000-12-31
DayOfWeek	Dia da semana. VARCHAR	Monday
Week	Semana da data respetiva. INT	2
Month	Mês da data respetiva. VARCHAR	June
Quarter	Trimestre da data respetiva. INT	2
YEAR	Ano da data respetiva.	2000

Tabela 5 - Descrição da dimensão calendário – dim_date.

Quanto ao segundo data mart, o data mart "Queries", este inclui na sua estrutura todas as dimensões do data mart anterior, contendo porém uma nova dimensão: a dimensão Query, que possui informação específica relativa às queries lançadas

pelos utilizadores. Esta dimensão permitirá estudar quais as dimensões envolvidas nas queries MDX que foram lançadas. O diagrama lógico deste segundo data mart está apresentado na figura 15.

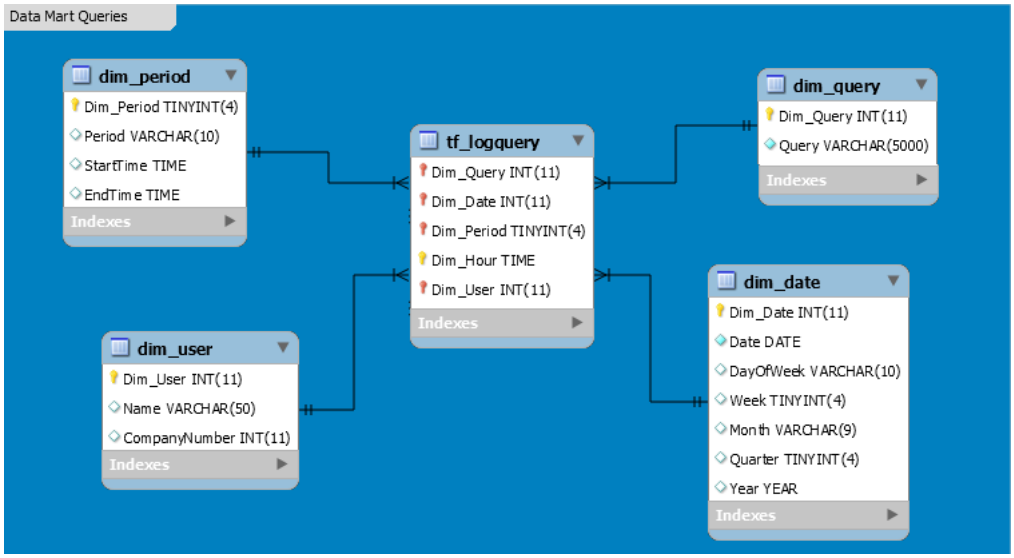


Figura 15 – Modelo lógico do data mart "Queries".

A descrição da tabela de factos e da dimensão query estão apresentadas, de seguida, nas tabelas 6 e 7.

Tabela de factos de queries: Contém informação relativa às sessões praticadas pelos utilizadores durante as respetivas sessões de exploração OLAP.		
Atributo	**Descrição/Tipo**	**Exemplo**
Dim_Query (FK)	Identificador da query. INT	1
Dim_Date (FK)	Data de lançamento da query. INT	1
Dim_Period (FK)	Período de lançamento da query. INT	1
Dim_Hour	Hora de lançamento da query. TIME	10:00:00
Dim_User (FK)	Utilizador que lançou a query. INT	José Mourinho

Tabela 6 - Descrição da tabela de factos de queries - "tf_logquery".

Dimensão Query: Contém informação relativa às queries MDX lançadas pelos utilizadores.		
Atributo	**Descrição/Tipo**	**Exemplo**
Dim_Query (PK)	Identificador do query. INT	1

Query	Expressão da query. Varchar	SELECT ...

Tabela 7 - Descrição da dimensão query – "dim_query"

Os processos de sessões que alimentam o programa desenvolvido (ver secção seguinte) obtêm os dados acerca das sessões a partir do data mart "Sessões".Para isso, fazem uma query para determinar a hora de início e de fim de cada uma das sessões associadas a um determinado utilizador, relativamente a um determinado dia e a um determinado período, se esse for o caso. De seguida, lançam uma segunda query ao data mart "Queries" de forma a obter todas as queries associadas ao utilizador em questão e que foram lançadas nos períodos de tempo determinados pela query submetida anteriormente, por ordem cronológica, de forma a estruturar as queries por tempo, da mais antiga à mais recente. Isto é imprescindível para o cálculo da cadeia de Markov do algoritmo de esfoliação proposto neste trabalho de dissertação. Como é possível ver através dos dois diagramas lógicos apresentados para os data marts (figura 16), as dimensões "Utilizador", "Período" e "Calendário" são dimensões partilhadas entre os dois data marts.

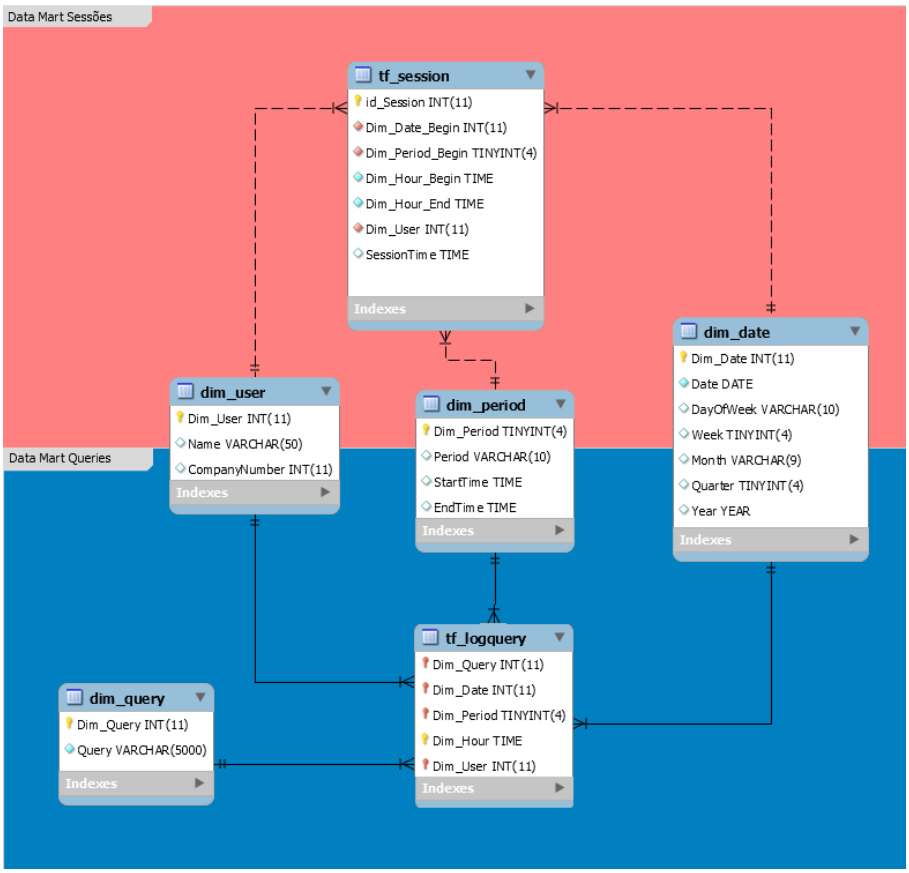


Figura 16 – Vista geral dos Data Marts com dimensões partilhadas

As queries MDX, que estão presentes na dimensão "Query", são relativas a um hipercubo criado e povoado a partir da base de dados "FoodMart", que é uma base de dados baseada na Microsoft Resources (2015). Esta base de dados é apresentada como um data mart, com configuração de esquema em estrela, contendo as medidas "Store Sales", "Store Costs" e "Unit Sales", e as dimensões "Product", "Time", "Customer", "Store" e "Promotion" (figura 17). A "FoodMart" tem como objectivo moldar uma perspetiva de análise de vendas e custos de uma cadeia de lojas e as promoções efetuadas em cada um das vendas efetuadas. Será sobre estes dados que o programa desenvolvido aplicará o algoritmo de esfoliação de hipercubos concebido nesta dissertação.

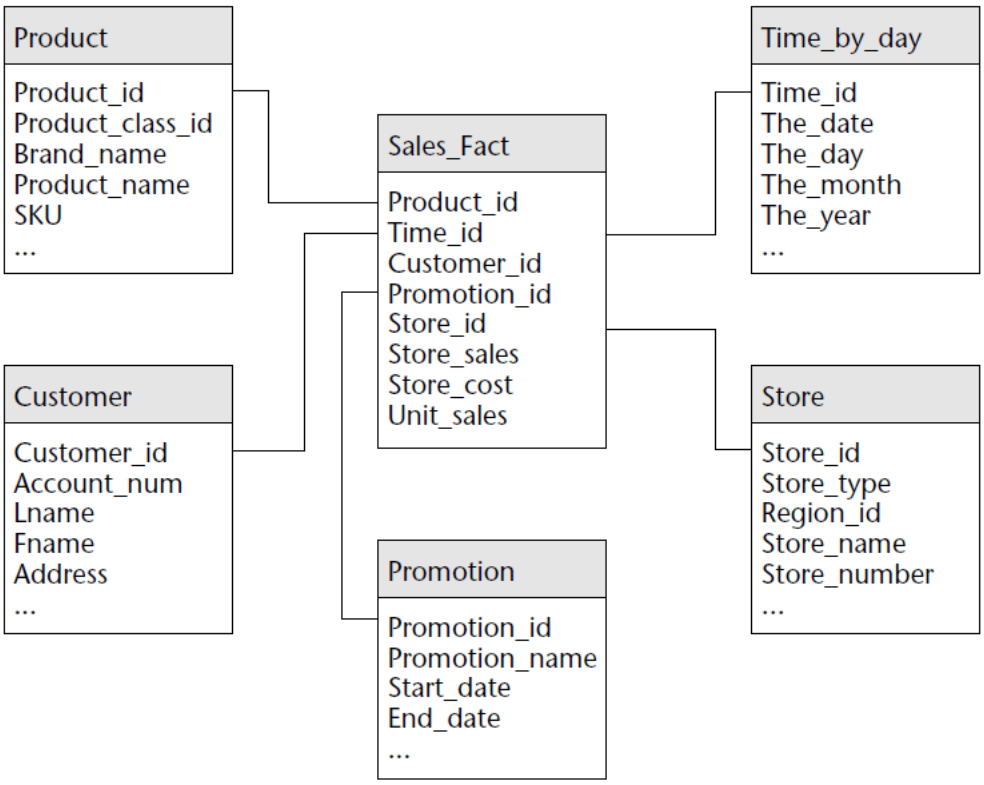


Figura 17 – O esquema-estrela da base de dados "FoodMart".

4.3 Aplicação desenvolvida

Tal como referido no início deste capítulo, a aplicação idealizada foi desenvolvida através da linguagem Java. A aplicação tem como objetivo recriar o algoritmo que foi anteriormente proposto nesta dissertação. Para isso foi necessário utilizar várias bibliotecas específicas, como a bem conhecida *swing,* para criar a interface gráfica e *a graphstream*, para criar a cadeia de Markov. Esta última foi escolhida porque, para além de ser *open source* e ter uma forte documentação exposta, é uma biblioteca que serve para modelar e analisar gráficos dinâmicos, sendo possível com ela gerar, importar e exportar gráficos bem como visualizá-los (GraphStream, 2015).

Vejamos então como é que esta aplicação está organizada. Na figura 18 vemos o estado da aplicação no momento em que é aberta. Nesta situação, a sua interface está organizada em quatro secções distintas. Na parte superior, pode-se ver incorporado um painel com o nome Markov Chain, que tem como objetivo

apresentar a cadeia de Markov para a configuração presente na secção de preferências e um outro painel que tem como objetivo apresentar as classes de equivalência das dimensões e das medidas. O gráfico da cadeia foi gerado com o auxílio da biblioteca *streamgraph* e integrado na estrutura *swing* na construção da interface gráfica.

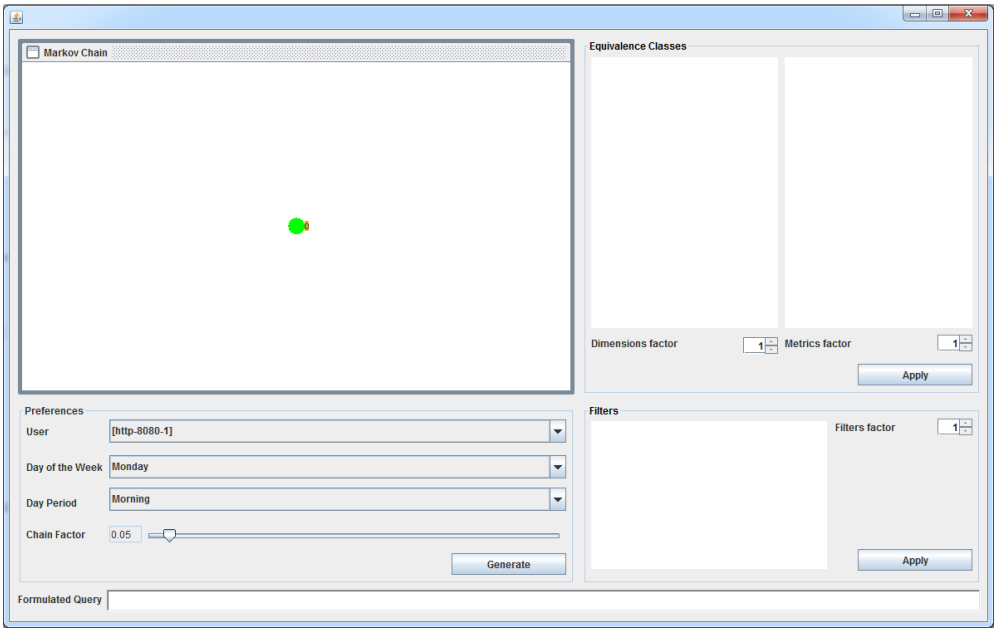


Figura 18 – Screeshot da aplicação no momento da sua abertura.

Na primeira figura anteriormente apresentada podemos identificar na cadeia de Markov, para a sua configuração inicial, que esta apenas contém o estado inicial. É de notar que o estado inicial é identificado por um nodo de cor verde e o estado final por um nodo de cor vermelha – este porém não se encontra visível na figura. Isso deve-se ao facto de não existirem sessões para aquele utilizador, para o dia e período em questão, ou então é porque simplesmente não foi possível gerar uma cadeia de Markov que respeitasse o fator de probabilidade inicial. Neste caso, esse fator tem o valor de 5%. Isso justifica o facto de o painel da secção das classes de equivalência estar vazio e a restante parte da interface também – veremos este caso já de seguida.

No painel localizado na parte inferior do ambiente apresentado na figura 18 encontram-se os dois outros paineis que têm o nome de Preferences e Filters. O primeiro tem como objetivo proporcionar ao utilizador a oportunidade de

definir a configuração mais adequada para gerar a query que representará a vista a materializar. Neste painel é possível alterar qual o utilizador, o dia da semana e o período do dia para o qual a aplicação deve preparar e apresentar os resultados. Na figura 19 podemos ver esta secção do painel de preferências.

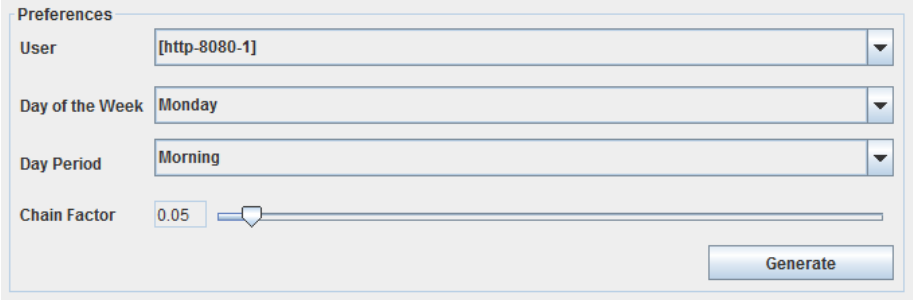


Figura 19 - Secção da aplicação para mudar a sua configuração base.

Os três campos que se observam nesse painel surgem na forma de *combo box,* sendo a informação aí presente retirada diretamente da base de dados e preenchida de forma dinâmica. O slider localizado mais abaixo, com a legenda Chain Factor, tem como funcionalidade alterar o *threshold* inferior para o qual a cadeia de Markov deve apresentar o gráfico com os arcos superiores a essa factor de probabilidade, isto é, o grafo de probabilidade acima desse valor. A secção à direita, e última, tem como finalidade apresentar os filtros para as queries que se encontram na cadeia de Markov.

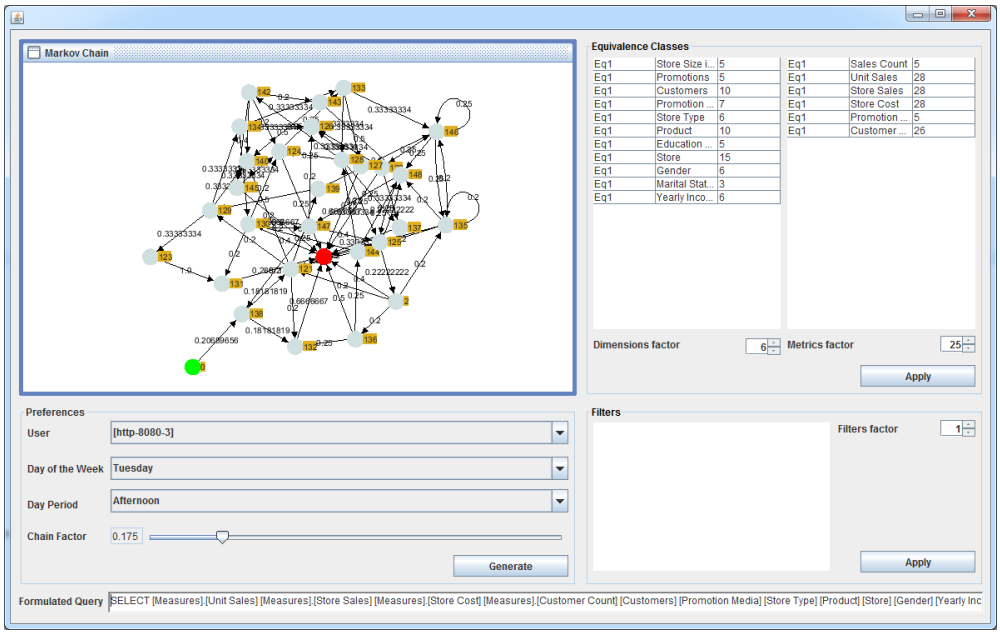


Figura 20 – Configuração da aplicação para um dado utilizador

Mudando a configuração no painel das preferências para uma configuração que apresente dados potencialmente relevantes, alcançamos o estado que a figura 20 apresenta. Esta figura mostra a configuração para um determinado utilizador, dia e período do dia, no qual o fator de probabilidade da cadeia de Markov é de 17.5% e os fatores de dimensão e métricas que, neste caso, são de 6 e 25, respetivamente. Desta forma são apresentadas e construídas a cadeia de Markov e respetivas classes de equivalência. O que resulta na geração da seguinte query:

```
SELECT [Measures].[Unit Sales] [Measures].[Store Sales]
[Measures].[Store Cost] [Measures].[Customer Count] [Customers]
[Promotion Media] [Store Type] [Product] [Store] [Gender] [Yearly
Income] FROM cube
```

que representa o cubo esfoliado a ser materializado:. De forma a dar uma melhor percepção da interface, as secções serão apresentadas isoladamente nas próximas figuras.

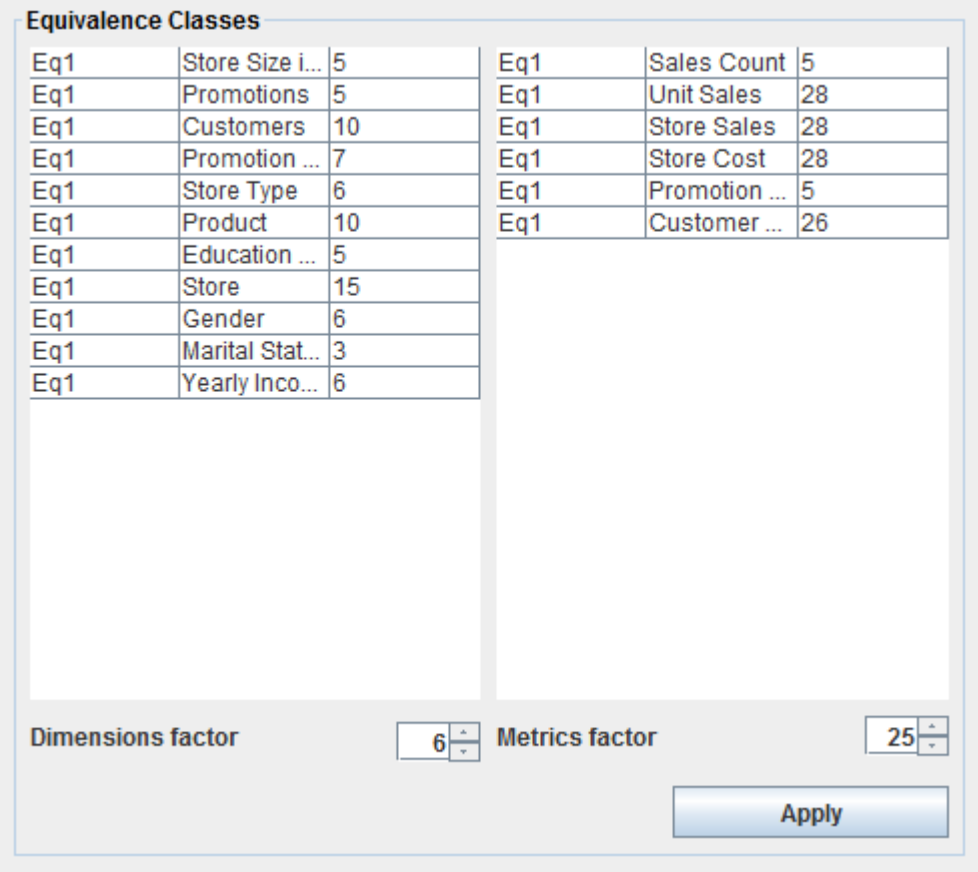


Figura 21 - Painel relativo às classes de equivalência

Os *spinners* com as legendas Dimensions factor e Metrics factor têm como finalidade suportar a escolha do threshold inferior para as quais as dimensões e métricas identificadas farão parte da query gerada. O botão Apply permite aplicar as configurações atribuídas aos *spinners* de forma a serem refletidas na query.

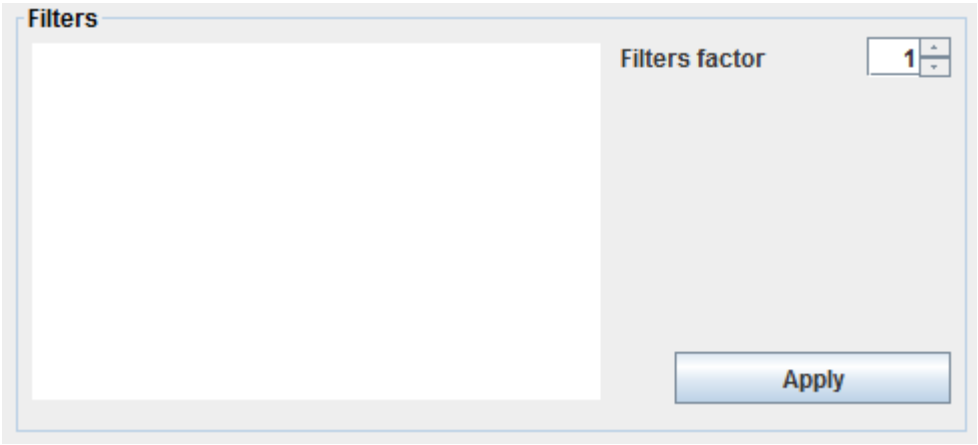


Figura 22 - Painel relativo aos filtros

O painel relativo aos filtros a aplicar não se encontra preenchido, pois as queries que fazem parte da cadeia de markov para a configuração dada não têm filtros na sua definição. No entanto, este quadro apresenta a junção dos filtros de todas as queries da cadeia, num formato igual ao das classes de equivalência, em que a primeira coluna identifica a dimensão, a segunda o filtro em si e a terceira o factor de incidência do filtro, isto é, quantas vezes aquele filtro aparece.

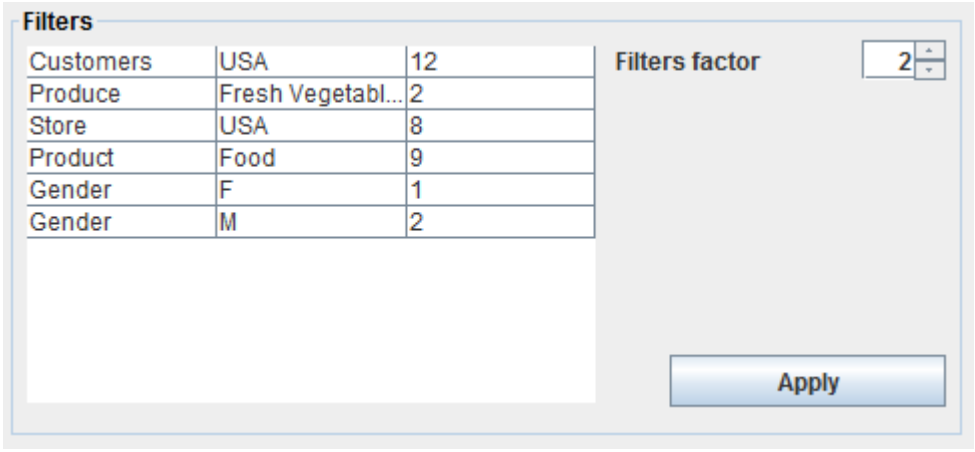


Figura 23 - Painel de filtros para uma dada configuração.

Na figura 24 é possível identificar claramente o nodo do estado inicial, a verde, e o nodo do estado final, a vermelho. Para os dados existentes na base de dados, a geração dos resultados por parte da aplicação é praticamente instantânea, o que revela uma boa ferramenta de suporte, com uma interface intuitiva e com uma apresentação de dados bastante agradável.

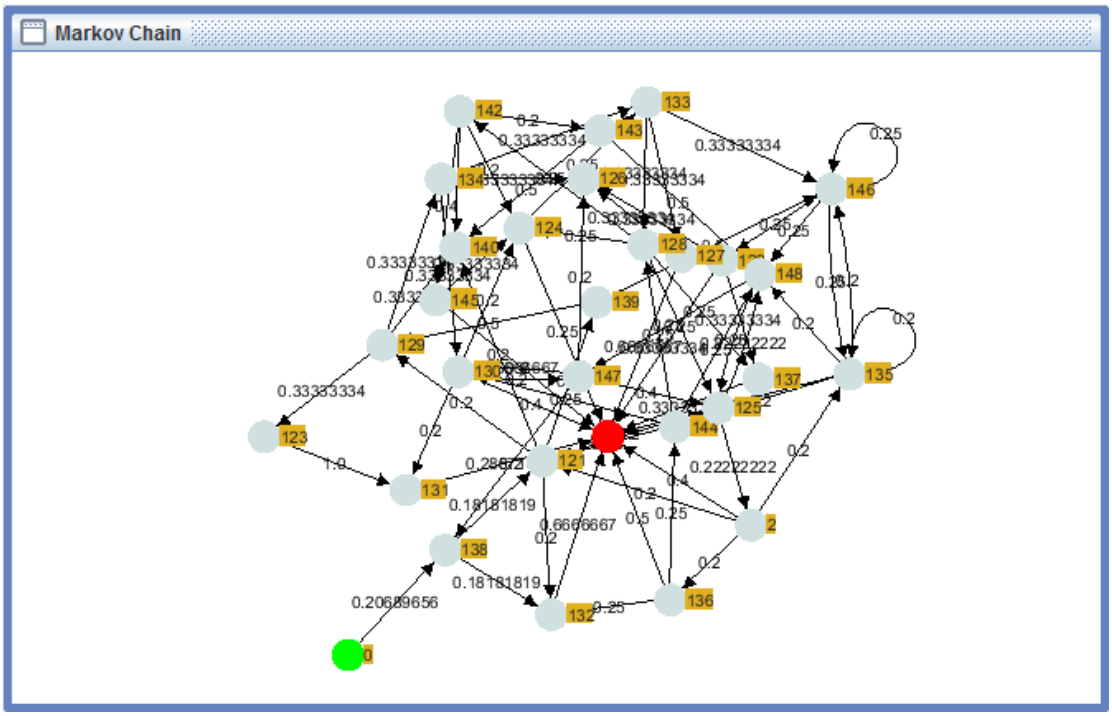


Figura 24 - Cadeira de markov da aplicação para a nova configuração

4.4 Análise dos Resultados

Como vimos na secção anterior, a ferramenta desenvolvida ajuda o utilizador a escolher qual a estrutura para um determinado hipercubo através de uma query que é sugerida e que representa uma vista da parte do hipercubo a materializar. Isto é feito com base na informação que o sistema tem acerca das sessões e das queries previamente lançadas pelo utilizador durante as suas sessões de exploração OLAP, referentes a um determinado período temporal.

A ferramenta desenvolvida pode ser alimentada gradualmente com novos dados, uma vez que o seu trabalho assenta na definição das sessões OLAP estabelecidas, o que permite que nova informação sobre novas sessões e sobre novas queries possam ir sendo adicionadas, bem como outra informação relacionada com novos utilizadores.

Durante a realização do trabalho de dissertação, verificámos que o método das classes de equivalência de Niemi (2001), por si só, não é muito relevante no que diz respeito a sessões OLAP e a grandes conjuntos de queries, uma vez que tem

tendência a criar apenas uma classe de equivalência. Isto pode ser justificado pelo seguinte:

1) uma sessão OLAP é caracterizada por ser uma sequência de queries relacionadas entre si, o que significa que é muito provável que todas elas façam parte da mesma classe de equivalência.
2) quando o conjunto de queries que fazem parte da cadeia de Markov for consideravelmente grande, as queries tem tendência a pertencer à mesma classe de equivalência, visto que podem partilhar uma ou mais dimensões entre si.

Estas duas razões fazem com que o método das classes de equivalência apenas identifique qual é o seu objetivo e as diferentes áreas de dados sobre as quais as queries operaram. Nas figras 25 e 26 podemos observar duas cadeias de Markov distintas e as respetivas classes de equivalência que foram geradas em associação com elas.

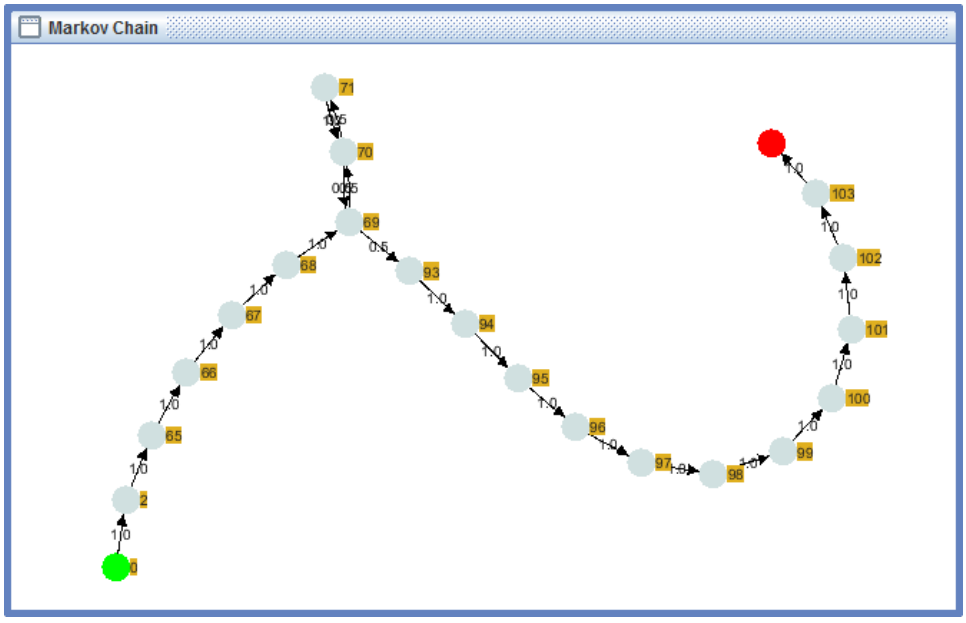


Figura 25 - Cadeia de Markov 1

A cadeia de Markov apresentada na figura 25 representa as sessões de um utilizador para um dado dia e para um dado período. Esta cadeia apresenta um comportamento bem determinado relativo à exploração realizada por um utilizador. Nela podem-se verificar que as probabilidades de passagem entre as diferentes queries são muito altas, sendo a mínima observável de 50%.

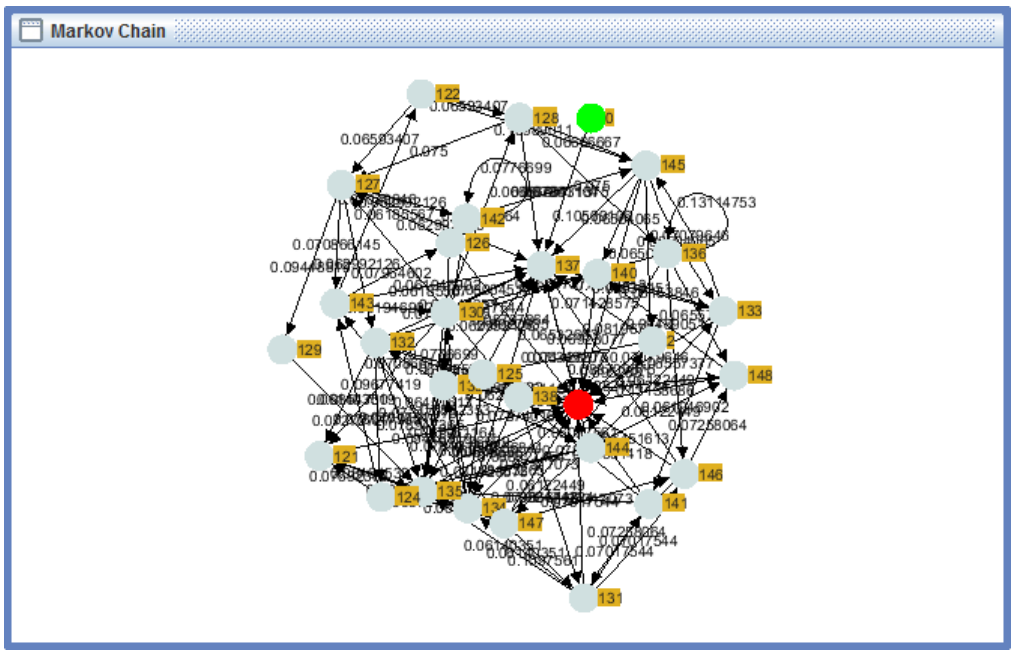


Figura 26 – A cadeia de Markov 2.

Por sua vez, esta segunda cadeia apresenta um padrão de exploração mais diversificado e integra um conjunto de queries mais alargado. Tal como era expectável, com base nos dados desta cadeia apenas uma classe de equivalência é obtida para cada um dos casos, tal como pode ser observado através dos dados das figuras 27 e 28.

Equivalence Classes

Eq1	Customers	20	Eq1	Sales Count	11
Eq1	Promotion ...	1	Eq1	Unit Sales	34
Eq1	Produce	3	Eq1	Store Sales	19
Eq1	Store	21	Eq1	Store Cost	19
Eq1	Product	55	Eq1	Promotion ...	11
Eq1	Education ...	7	Eq1	Customer ...	11
Eq1	WA	1			
Eq1	Gender	20			
Eq1	Marital Stat...	5			

Figura 27 – As classes de equivalência para a cadeia 1

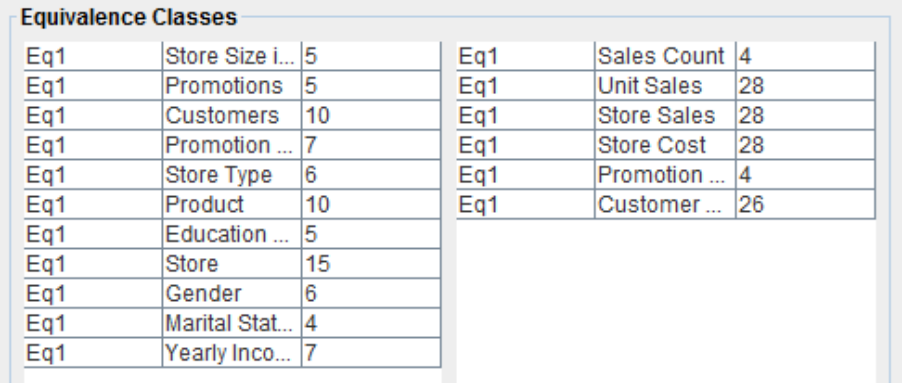


Figura 28 – As classes de equivalência para a cadeia 2.

A identificação de uma classe de equivalência é importante na determinação da zona de atuação de um determinado conjunto de queries. Contudo quando a classe de equivalência traduz praticamente todas as dimensões existentes, essa informação torna-se, obviamente, irrelevante (ou quase). É, neste tipo de casos, que temos que utilizar as classes de equivalência. Como podemos ver através das figuras 27 e 28, relativas às classes de equivalência, o fator de incidência consegue desmarcar bem quais as dimensões que têm um maior impacto ou maior interesse para os utilizadores. Mesmo no caso da segunda cadeia, que apresenta um conjunto mais diversificado de relações de passagem entre queries, o método das classes de equivalência estendidas traz-nos alguns benifícios, apesar de não revelar grande discrepância entre as dimensões. O mesmo acontece no caso das medidas. Na maioria das vezes, apenas se consegue identificar uma classe de equivalência. Porém, mais uma vez, o fator de incidência ajuda a identificar de forma bastante clara, qual ou quais das medidas são as mais relevantes.

Estando bem identificadas as dimensões mais relevantes, os filtros auxiliam ainda mais na filtragem de informação, de forma a tornar o processo de esfoliação ainda mais preciso. Muitas vezes estes filtros estão associados às dimensões mais importantes indentificadas pelo método de estensão. A figuras seguinte mostra os filtros obtidos para a primeira cadeia, visto que apenas esta tinha filtros associados às queries.

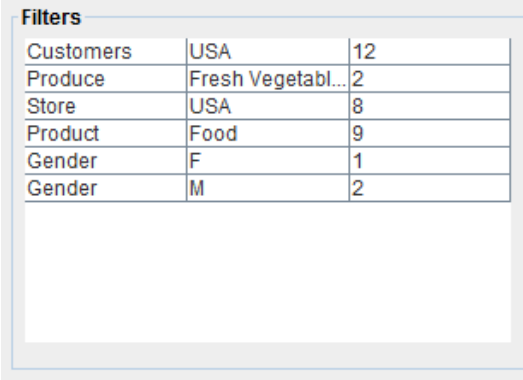


Figura 29 - Filtros selecionados para a cadeia 1.

É desta forma que a ferramenta auxilia a construção de uma query que representará a vista do cubo a ser materializada. O utilizador ao visualizar a cadeia de Markov, deve escolher o *threshold* de probabilidade máximo de forma a obter um conjunto razoável de queries, que não esteja muito distante daquele que é obtido usualmente sem a aplicação desse *threshold.* Isto faz com que se obtenha um conjunto de queries que são as que frequentemente são lançadas por esse utilizador e, assim, evitar construir classes de equivalência com dimensões que fazem parte de queries menos frequentes e que, por isso, obtenham um fator de incidência considerável ou que contribuam para o fator de incidência de outras dimensões, tornando-as mais importantes do que o são na realidade. De seguida, com a apresentação das classes de equivalência estendidas para a cadeia obtida, é possível perceber quais as dimensões com o maior foco de interesse. Aqui, cabe ao utilizador escolher um fator de incidência que permita "capturar" essas dimensões. Esta escolha pode ser mais ou menos fácil, consoante as discrepâncias encontradas no cálculo do fator de incidência. Da mesma forma, o utilizador deve perceber quais as medidas mais importantes a aplicar e atuar de forma análoga. Assim, para o primeiro caso, são facilmente identificadas as dimensões "Customers", "Store", "Product" e "Gender", no caso das medidas são as "Unit Sales", "Store Sales" e "Store Cost". Com os filtros associados a estas dimensões, a query sugerida seria algo do género:

```
SELECT
[Measures].[Unit Sales], [Measures].[Store Sales], [Measures].[Store
Cost], [Customers], [Store], [Product], [Gender] FROM cube
WHERE [Customers].[USA] and [Store].[USA] and [Product].[Food]
```

No caso da segunda cadeia de Markov a escolha é um pouco mais difícil, visto que os fatores de incidência das dimensões é mais equilibrado. Para este caso as dimensões mais solicitadas são as dimensões "Customers", "Product" e "Store", o que geraria a query:

```
SELECT [Customers], [Product], [Store] FROM cube
```

É esperado que, neste caso, a vista a materializar não seja tão eficaz como a do caso anterior, uma vez que as dimensões mais relevantes não são aqui tão evidentes nem que existem filtros associados às queries. Isto pode fazer com que apareçam mais queries que não sejam satisfeitas pela vista a gerar e como tal seja necessário recorrer à fonte de dados para que uma resposta possa ser fornecida.

Com esta ferramenta é possível, também, conhecer quais as estruturas dos cubos disponíveis que provavelmente irão ser utilizados durante a próxima semana. Na tabela 8 podemos observar alguns exemplos de queries que foram geradas durante a exploração da aplicação desenvolvida, com a finalidade de sugerir a estrutura dos cubos a materializar num dado período no futuro.

Especificações	**Query gerada**
Utilizador: [Http-8080-1] **Dia da Semana**: Monday **Período do dia:** Afternoon **Grau de confiança da cadeia:** 95% **Fator Dimensões:** 25 **Fator Medidas:** 13 **Fator Filtros:** 5	`SELECT [Measures].[Sales Count] [Measures].[Store Sales] [Measures].[Unit Sales] [Measures].[Customer Count] [Customers] [Store] [Product] FROM cube WHERE [Product].[Food] AND [Store].[USA]`
Utilizador: [Http-8080-1] **Dia da Semana**: Tuesday **Período do dia:** Afternoon	`SELECT [Measures].[Sales Count] [Measures].[Store Sales] [Measures].[Unit Sales] [Measures].[Customer Count]`

Grau de confiança da cadeia: 100% **Fator Dimensões:** 25 **Fator Medidas:** 13 **Fator Filtros:** 5	`[Customers] [Store] [Product] FROM cube WHERE [Product].[Food] AND [Store].[USA]`
Utilizador: [Http-8080-1] **Dia da Semana**: Wednesday **Período do dia:** Afternoon **Grau de confiança da cadeia:** 50% **Fator Dimensões:** 20 **Fator Medidas:** 10 **Fator Filtros:** 5	`SELECT [Measures].[Sales Count] [Measures].[Store Sales] [Measures].[Unit Sales] [Measures].[Customer Count] [Customers] [Store] [Product] FROM cube WHERE [Customers].[USA] AND [Product].[Food] AND [Store].[USA]`
Utilizador: [Http-8080-1] **Dia da Semana**: Wednesday **Período do dia:** Morning **Grau de confiança da cadeia:** 5.2% **Fator Dimensões:** 20 **Fator Medidas:** 15 **Fator Filtros:** 10	`SELECT [Measures].[Sales Count] [Measures].[Unit Sales] [Measures].[Store Sales] [Measures].[Customer Count] [Customers] [Product] [Store] FROM cube WHERE [Customers].[USA] AND [Product].[Food]`
Utilizador: [Http-8080-2] **Dia da Semana**: Monday **Período do dia:** Night **Grau de confiança da cadeia:** 30% **Fator Dimensões:** 20 **Fator Medidas:** 20 **Fator Filtros:** 3	`SELECT [Measures].[Unit Sales] [Customers] [Store] [Product] [Gender] FROM cube WHERE [Customers].[USA] AND [Store].[USA] AND [Product].[Food]`
Utilizador: [Http-8080-2] **Dia da Semana**: Monday **Período do dia:** Night **Grau de confiança da cadeia:** 35% **Fator Dimensões:** 18	`SELECT [Measures].[Unit Sales] [Measures].[Store Sales] [Measures].[Store Cost] [Promotion Media] [Product] FROM cube`

Fator Medidas: 5 **Fator Filtros:** 1	
Utilizador: [Http-8080-3] **Dia da Semana**: Monday **Período do dia:** Morning **Grau de confiança da cadeia:** 5.5% **Fator Dimensões:** 6 **Fator Medidas:** 10 **Fator Filtros:** 10	`SELECT [Measures].[Unit Sales] [Measures].[Store Sales] [Measures].[Store Cost] [Measures].[Customer Count] [Customers] [Promotion Media] [Store Type] [Store] [Product] [Gender] [Yearly Income] FROM cube`
Utilizador: [Http-8080-3] **Dia da Semana**: Monday **Período do dia:** Morning **Grau de confiança da cadeia:** 10% **Fator Dimensões:** 6 **Fator Medidas:** 10 **Fator Filtros:** 10	`SELECT [Measures].[Unit Sales] [Measures].[Store Sales] [Measures].[Store Cost] [Measures].[Customer Count] [Store Size in SQFT] [Promotions] [Customers] [Promotion Media] [Store Type] [Product] [Education Level] [Store] [Gender] [Yearly Income] FROM cube`

Tabela 8 - Tabela de previsão de cubos a materializar no futuro

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões

As ferramentas OLAP tem como objetivo proporcionar aos utilizadores tempos de resposta aceitáveis independentemente da complexidade da query ou do tamanho dos dados que albergam. Contudo, estas mesmas ferramentas podem ser dispendiosas em termos de recursos que utilizam para satisfazer tal compromisso. O método desenvolvido no âmbito desta dissertação tenta então trazer o melhor destes dois mundos, isto é, sugerir uma vista do hipercubo a materializar, sendo esta vista particularmente especial no sentido em que a maioria das queries devem incidir sobre esta e dessa forma reduzir os recursos necessários.

O método desenvolvido pode ser visto também como um algoritmo para ações de *caching*, que pode ser usado para criar uma estrutura de dados de primeira linha de acesso para uma aplicação. Isto, porque, simplesmente tira partido da informação relativa às sessões de exploração dos utilizadores para sugerir uma vista a materializar, vista esta que deve conter os dados mais solicitados durante as análises. As sessões permitem retirar o perfil de utilização dos utilizadores e construir uma cadeia de Markov que traduz as relações de precedência entre queries e respetivas probabilidades. Daqui advém todo o poder de um sistema de monitorização de sessões, pois tais relações permitem descobrir qual o foco de interesse dos utilizadores e consequentemente descobrir quais os dados mais relevantes para estes. No entanto o método permite ainda ir um pouco mais além e, dentro dos dados mais relevantes, gerar hierarquias de relevância através do método de extensão de classes de equivalência, que atribuem graus de incidência

às dimensões e métricas que são analisadas. Por fim são adicionados filtros que mais uma vez esfoliam o cubo de dados e que dão mais enfâse ao objetivo de garantir uma vista de dados consolidada e mais pequena.

A problemática de perceber de que forma é possível diminuir o tamanho de hipercubos de forma a dar maior realce às necessidades dos utilizadores tem vindo a ser um tema de forte discussão por parte da comunidade científica ao longo dos últimos anos. O método aqui apresentado combina algumas das ideias que foram sendo apresentadas para solucionar este problema, como o caso das classes de equivalência de Niemi e das cadeias de Markov, tendo aplicado estes dois métodos na definição de sessões de exploração de cubos. Este método permite concluir que um estudo assente nas sessões de exploração OLAP para criar uma ponte entre os dados de um hipercubo e os dados que são efetivamente usados e analisados pelos utilizadores. A partir da análise de dados efetuada foi possível verificar que é possível muitas vezes retirar uma vista materializada, cujos dados poderão corresponder, com um bom grau de confiança, àqueles que os utilizadores estão interessados. Tal método pode ser usado de forma a aumentar o desempenho das ferramentas OLAP e de outras aplicações.

5.2 Trabalho futuro

Como foi possível constatar ao longo deste documento, as sessões OLAP podem ter um papel fundamental na descoberta dos dados mais relevantes para um determinado conjunto de utilizadores. Assim, a existência de um mecanismo de monitorização de sessões OLAP faz com que seja possível estudar possíveis formas de reestruturação de estruturas de dados, com o objetivo de as tornar menos dispendiosas em termos de recursos computacionais. Algumas melhorias mais imediatas podem ser identificadas nos parágrafos que se seguem.

A existência de uma análise das sessões dentro de um determinado período de tempo é de grande importância. Ora, um negócio não se rege sempre pelas mesmas regras, os mercados estão sempre em constante mudança e evolução, e as organizações devem saber mudar o seu rumo de forma a responder o mais

rapidamente possível a estas mudanças. Muitas vezes responder a estas mudanças significa mudar o foco de interesse dos próprios negócios e isto tem um impacto imediato na forma como os analistas estudam os dados. Desta forma, a ferramenta deve possuir um mecanismo que permite a filtragem das sessões para um determinado período de tempo de forma a estudar apenas as novas tendências de consulta e determinar a melhor vista a materializar dos dados. O motivo pelo qual a ferramenta desenvolvida ainda não apresenta esta funcionalidade, deve-se ao facto dos dados existentes na fonte de dados estarem dentro de uma janela temporal de uma semana. Caso esta funcionalidade não seja implementada de futuro os problemas que se levantam em termos do método formulado são:

- A Cadeia de Markov apresentaria novas queries relativas aos novos interesses de análise e as mais antigas entrariam também na rede. Isto faria com que as classes de equivalência fossem calculadas de forma errada.
- As classes de equivalência calculadas com os dados relativos a interesses diferentes de análise, fariam com que o fator de incidência das classes fosse enganoso, visto que as queries que não pertenceriam ao foco do analista entrassem e valorizassem determinadas dimensões.
- A query gerada que representaria a vista a materializar não teria de todo um grande grau de confiança.

A ferramenta deve também ajudar a perceber se os analistas mudaram de comportamento. Uma outra funcionalidade a implementar seria o suporte a dados relativos a diferentes cubos de dados. Esta funcionalidade seria facilmente implementada, pois apenas deve oferecer a opção de escolher qual o cubo de dados que se pretende que seja analisado e filtrar os dados referentes a esse cubo. Estas duas modificações seriam aquelas que a curto prazo poderiam tornar a ferramenta mais interessante e completa, mas que para os efeitos de investigação feita e pelos dados disponíveis não se mostraram relevantes, nem trariam diferenças aos resultados, visto que existem apenas sessões e queries relativas a um único cubo de dados e dentro de uma janela de uma semana.

Relativamente aos dados usados pela ferramenta, as queries envolvidas em processos de pesquisa dos utilizadores apresentam filtros relativamente simples, em que um dado atributo de uma dimensão é igualado a um dado valor. Isto é algo que deve ser estudado de forma mais detalhada no futuro. Isto é, pensar num processo que permita obter queries que apresentassem filtros algo mais complexos e perceber de que forma esses filtros poderiam ser trabalhados no sentido de evoluir a geração da query que representa a vista a materializar.

Por outro lado, também poderiam ser estudas melhorias a serem inseridas em cada um dos passos do método que foi proposto, tanto ao nível da cadeia de Markov, como ao nível do processo de extensão das classes de equivalência realizado neste trabalho de dissertação. Poderiam ser também analisadas outras técnicas para complementar o método proposto, de forma a torná-lo mais completo e preciso na geração da query da vista a materializar.

Referências

[Paulraj Ponniah 2001] “Data Warehousing Fundamentals” - A comprehensive Guide for IT Pro, 2001.

[A. Cuzzocrea & S. Mansmann] “OLAP visualization: models, issues, and techniques”, Encyclopedia of Data Warehousing and Mining, 2nd ed., IGI Global, Hershey, PA, USA, pp. 1439-1446, 2009.

[Connolly & Begg 2005] “Database Systems: A Practical Approach to Design, Implementation, and Management” (4nd Edition). Addison Wesley, 2005.

[Kimball & Ross 2002] “The Data Warehouse ToolKit: The Complete Guide to Dimensional Modeling”, 2002.

[Matteo Golfarelli & Stefano Rizzi 2009] “Data Warehouse Design Mordern Principles and Methodologies”. McGraw-Hill Osborne Media, 2009.

[Harinarayan et al. 1996] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. 1996. "Implementing data cubes efficiently". In Proceedings of the 1996 ACM SIGMOD international conference on Management of data (SIGMOD '96), Jennifer Widom (Ed.). ACM, New York, NY, USA, 205-216.

[Lin et al. 2007] Ziyu Lin, Dongqing Yang, Guojie Song, and Tengjiao Wang. 2007. “User-Oriented Materialized View Selection”. In Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT '07). IEEE Computer Society, Washington, DC, USA, 133-138.

[Cheung el al. 1999] David W. Cheung, Bo Zhou, Ben Kao, Hongjun Lu, Tak W. Lam, and Hing Fung Ting. “Requirement Based Data Cube Schema Design”. In proceedings of the Eighth International Conference on Information and

Knowledge Management : CIKM '99, Kansas City, MI, USA, ACM, New York, USA, 2-6 Nov. 1999, p. 162-169.

[Niemi et al. 2001] Niemi, T.., Nummenmma, J.. & Thanisch, P. 2001 “Constructing OLAP Cubes Based on Queries”. In Processidings of ACM DOLAP.

[Rocha & Belo 2015] Daniel Rocha and Orlando Belo. “Integrating usage analysis on cube view selection – an alternative method”. Int. J. Decision Support Systems, Vol. 1, No. 2, 2015.

[Borges et al. 1999] Borges, J. & Levene, M. “Data Mining and User Navigation Patterns”. San Diego, California, USA: Springer.

[Sarukkai, R.R. 2000] Sarukkai, R.R. 2000 . “Link prediction and path analysis using Markov chains”. Proceedings of the 9th international World Wide Web conference on Computer networks. Amsterdam, Holand, 377-386.

[Deshpande et al. 2004] Deshpande, M. & Karypis, G. 2004. “Selective Markov models for predicting Web page accesses”. ACM Trans. Internet Technol., vol. 4, 2, May, pp. 163-184.

[D. Boukraa et al.] Doulkifli Boukraa, Omar Boussaid, Fadila Bentayeb. “OLAP Operators for Complex Object Data Cubes”. High School of Computer Science, Oued-Smar, Algiers.

[Wil et al.] Wil M.P. van der Aalst. “Process Cubes: Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining”.

[Anna Rozeva 2007] Anna Rozeva “Dimensional Hierarchies – Implementation in Data Warehouse Logical Scheme Design”. International Conference on Computer Systems and Technologies – CompSysTech 2007.

[Pedersen et al. 2001] Tobern Bach Pedersen and Christian S. Jensen "Multidimensional database technology". IEEE Computer, 40 – 46, 2001.

[Gray el al. 1995] Jim Gray, Adam Bosworth, Andrew Layman and Hamid Pirahesh. "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals". Technical Report. MSR-TR-95-22.

[Codd et al. 1993] E.F Codd, S.B Codd and C.T Salley 1993. "Providing OLAP to User-Analysts: An IT Mandate".

[Raymond el al. 1992] Darrell R. Raymond and Frank Wm. Tompa. "Applying Database Dependency Theory to Software Engineering". Department of Computer Science, University of Waterloo, N2L 3G1.

[Laurence el al. 2006] Michael Laurence and Andrew Chaplin. "Dynamic view selection for OLAP". 8th Int'l Conference on Data Warehousing

[Niemi el al. 2000] Niemi, T. Nummenmaa, J. and Thanisch, P. "Applying dependency theory to conceptual modelling", Topics in Conceptual Analysis and Modeling, Czech Academy of Sciences. Publishing House Filosofia, Prague, 2000.

[Matteo Golfarelli & Stefano Rizzi 2009] Matteo Golfarelli and Stefano Rizzi. "Expressing OLAP preferences". 21st International Conference on Scientific and Statistical Database Management, SSDBM 2009, Berlin.

[Chomicki 2003] Jan Chomicki. "Preference Formulas in Relational Queries". University of Buffalo, Buffalo, New York. P1: GSY.

[Koutrika el al. 2008] G. Koutrika and Y. Ioannidis. "Answering queries based on preferences hierarchies". In: Proc. VLDB, Auckland, New Zealand 2008.

Wisdomjobs. (2015). Diponível em: http://www.wisdomjobs.com/e-university/data-mining-tutorial-199/mining-olap-cubes-387.html. [Último acesso a: 01-08-2015].

GraphStream. (2015) *A Dynamic Graph Library.* Diponível em: http://graphstream-project.org/. [Último acesso a: 06-08-2015].

Athena. (2015) *Online Analytical Processing*. Diponível em: http://athena.ecs.csus.edu/~olap/olap/OLAPoperations.php. [Último acesso a: 10-08-2015].

Lemire. (2010) *Data Warehousing and OLAP*. Diponível em: http://lemire.me/OLAP/index.html. [Último acesso a: 11-08-2015].

Olap. (2015) *Codd's Paper.* Diponível em: http://olap.com/learn-bi-olap/codds-paper/. [Último acesso a: 09-07-2015].

Olap House. (2015) *A company of OLAP Experts*. Diponível em: http://www.olaphouse.com/. [Último acesso a: 27-08-2015].

Oracle. (2015) *Data Warehousing and Big Data.* Diponível em: http://www.oracle.com/technetwork/database/bi-datawarehousing/overview/index.html. [Último acesso a: 10-08-2015].

TechTarget. (2015) *OLAP Definition* Diponível em: http://searchdatamanagement.techtarget.com/definition/OLAP. [Último acesso a 29-08-2015].

Microsoft. (2015) *Overview Of Online Analytical Processing*. Diponível em: https://support.office.com/en-in/article/Overview-of-Online-Analytical-Processing-OLAP-15d2cdde-f70b-4277-b009-ed732b75fdd6. [Último acesso a: 28-08-2015].

Microsoft System Center. (2015) *Understanding OLAP Cubes.* Diponível em: https://technet.microsoft.com/en-us/library/hh916543.aspx. [Último acesso a: 28-08-2015].

Microsoft TechNet. (2015) *Data Warehousing and OLAP.* Diponível em: https://technet.microsoft.com/en-us/library/aa197903(v=sql.80).aspx. [Último acesso a: 28-08-2015].

Solver. (2015) *Data Warehouse vs OLAP Cube*. Diponível em: http://www.solverusa.com/blog/2014/04/data-warehouse-vs-olap-cube/. [Último acesso a: 30-08-2015].

Microsoft Resources. (2015) *Common Operations and Examples.* Disponível em: https://technet.microsoft.com/en-us/library/aa936684(v=sql.80).aspx. [Último acesso a: 30-08-2015].