Escola de Ciências e Tecnologia
Universidade de Évora

OLAP em âmbito hospitalar: Transformação de dados de enfermagem para análise multidimensional

João Miguel Fernandes Pereira da Silva

Orientador: José Saias

Dissertação submetida à Universidade de Évora
para obtenção do grau de Mestre em Engenharia Informática

Janeiro de 2012

Escola de Ciências e Tecnologia
Universidade de Évora

OLAP em âmbito hospitalar: Transformação de dados de enfermagem para análise multidimensional

João Miguel Fernandes Pereira da Silva

Orientador: José Saias

Dissertação submetida à Universidade de Évora
para obtenção do grau de Mestre em Engenharia Informática

Janeiro de 2012

Sumário

O processo de tomada de decisões há muito que deixou de se basear meramente em conhecimentos administrativos e experiências vividas pelos intervenientes directos no quotidiano das organizações. Com o sucessivo aumento da quantidade de dados que vão sendo angariados, tornou-se imperativo a utilização de métodos e ferramentas de apoio à decisão, de modo a realizar análises inteligentes e retirando destas, informações importantes que facilmente estariam camufladas. Um destes métodos de análise é designado por OLAP (*Online Analytical Processing*). As ferramentas OLAP permitem a análise de dados quantificáveis de diversas perspectivas, uma vez que operam sobre um repositório multidimensional, normalmente englobado na arquitectura de um *Data Warehouse*.
O objectivo principal da presente dissertação é descrever a construção de uma solução de OLAP para análise e extracção de informação de registos de enfermagem, com ênfase no desenvolvimento de um repositório de dados multidimensional, indispensável ao bom funcionamento deste tipo de ferramentas. Para além disso, é também descrito o restante processo de construção, sendo mostrada a solução final de OLAP.
Para concretização do objectivo descrito foram usadas diversas ferramentas *open source* desenvolvidas pela Pentaho, empresa dedicada ao desenvolvimento de software, para a temática de *Business Intelligence* onde se englobam as ferramentas de OLAP.

OLAP in hospital scope: Transforming nursing data for multidimensional analysis

Abstract

The process of decision making has long ceased to be based merely on the administrative knowledge and in the experiences of its actors in the daily organizations life. With the successive increase in the amount of data that are being raised, it became imperative to use methods and tools for decision support in order to carry out intelligent analysis and to collect from it important information that would be easily camouflaged. One of these methods of analysis is called OLAP (Online Analytical Processing). OLAP tools enable analysis of quantifiable data from multiple perspectives, since they operate on a multidimensional repository, usually enclosed in a data warehouse architecture.

The main objective of this thesis is to describe the construction of an OLAP solution for analysis and extraction of information on nursing records, with emphasis on developing a repository of multidimensional data, key to the good functioning of such tools. In addition, we describe the rest of the construction process, showing the final OLAP solution.

To achieve the above described objective, there have been used several open source tools developed by Pentaho, a company dedicated to software development on the subject of Business Intelligence, which include OLAP tools.

Prefácio

Este documento contém uma dissertação intitulada "OLAP em âmbito hospitalar: Transformação de dados de enfermagem para análise multidimensional", um trabalho do aluno *João Miguel Fernandes Pereira da Silva*, estudante de Mestrado em Engenharia Informática na Universidade de Évora, integrado num projecto proposto pela empresa *Hewlett-Packard*(HP).

O orientador deste trabalho é o Professor Doutor José Saias[1], do Departamento de Informática da Universidade de Évora.

O autor do trabalho é licenciado em Engenharia Informática, pela Universidade de Évora. A presente dissertação foi entregue em Janeiro de 2012.

[1] jsaias@di.uevora.pt

Agradecimentos

Gostaria de agradecer em primeiro lugar ao professor José Saias que foi incansável na sua orientação ao longo de todo o processo de desenvolvimento desta dissertação, estando sempre disponível para tirar qualquer dúvida ou efectuando sugestões e recomendações, indicando-me sempre o caminho certo, sendo que seria impossível a concretização deste trabalho sem a sua preciosa ajuda.

Um agradecimento especial ao meu Pai, à minha Mãe e à minha namorada Patrícia, por todo o apoio, carinho e motivação que me deram ao longo de todo o meu percurso académico, nunca deixando de acreditar nas minhas capacidades.

Por último gostaria de agradecer aos restantes membros da minha família, aos meus colegas e amigos que me ajudaram igualmente de alguma forma a concretizar este percurso.

Acrónimos

BI *Business Intelligence*
Temática que envolve um conjunto de componentes e técnicas, de modo a fornecer ao utilizador final ferramentas que lhe permitam angariar conhecimentos que facilitem o processo de tomada de decisões. Conceito por vezes traduzido para Inteligência Competitiva ou Inteligência Empresarial.

DW *Data Warehouse*
Estrutura onde são armazenados um grande conjunto de dados, com o intuito de fornecer suporte a sistemas de apoio à decisão.

ETL *Extraction, Transformation Loading*
Processo que se divide em três fases:

- Extraction: leitura dos dados de uma ou mais fontes.
- Transformation: conversão dos dados extraídos para que sejam correctamente carregados na base de dados final.
- Loading: carregamento dos dados para a base de dados final.

MDX *Multidimensional Expressions*
Linguagem de consulta de dados desenhada exclusivamente para suportar bases de dados de sistemas OLAP.

MOLAP *Multidimensional OLAP*
Tipo de OLAP cujo armazenamento dos dados é efectuado em estruturas especiais, por exemplo arrays.

OLAP *Online Analytical Processing*

Sistemas que fornecem um conjunto de operações especiais para manipular e analisar um grande volume de dados, normalmente contidos num DW.

OLTP *Online Transactional Processing*

Conjunto de actividades e sistemas associados ao registo de operações de transacção, numa base de dados de uma determinada organização.

ROLAP *Relational OLAP*

Tipo de OLAP cujo armazenamento dos dados é efectuado numa base de dados relacional.

SGBD *Sistema de Gestão de Base de dados*

Sistema cujo o objectivo é realizar a gestão e a correcta manutenção dos dados armazenados em bases de dados.

SQL *Structured Query Language*

Linguagem padrão de consultas para efectuar pedidos de informação a bases de dados.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Neste capítulo pretende-se descrever o contexto em que está inserido o trabalho realizado nesta dissertação. Poderão ser mencionados alguns conceitos cuja descrição, mais detalhada, será efectuada no capítulo seguinte.

Numa primeira fase é efectuado o enquadramento geral através da secção 1.1. Posteriormente, na secção 1.2, é descrita a motivação, os objectivos propostos para este trabalho. Por fim, na secção 1.3, é feito um resumo da restante estrutura da dissertação.

1.1 Enquadramento

A tomada de decisões sempre esteve presente na vida do ser humano, pois trata-se de uma necessidade imperativa e à qual é impossível escapar. Isto leva-nos obviamente a pensar, dado a procura incessante de aperfeiçoamento dos conhecimentos por parte do ser humano, sobre o que podemos fazer para melhorar a nossa capacidade para tomar decisões. Com isto chegamos facilmente à conclusão que quanto mais e melhor for a informação que temos ao nosso dispor, mais fácil se torna a tomada de decisões sobre um determinado tema.

Muito antes da era das Tecnologias de Informação (TI), em civilizações com culturas já

bastante evoluídas, como os Egípcios ou os Persas, entre outros, a utilização de métodos considerados de *Business Intelligence (BI)*[1] já era uma realidade, muito embora fossem métodos algo rudimentares. Estes angariavam informações vindas da natureza, como a posição dos astros, o comportamento das marés, etc., para assim melhorarem o seu nível de conhecimento e deste modo conseguirem mais facilmente tomar decisões ditas "inteligentes".

Com a chegada das TI, e desde os anos setenta, que a pesquisa e desenvolvimento dos sistemas de base de dados permitiu que estes passassem de simples sistemas de processamento para sofisticadas bases de dados relacionais. Através desta evolução, tomou forma a possibilidade de armazenamento de diversos tipos de dados, tais como imagens e documentos, entre outros tipos mais complexos, em estruturas de tabelas relacionais [Tam, 1998]. Com esta capacidade disponível foi possível às empresas começarem a armazenar todo o tipo de informação importante, como dados sobre clientes, inventários, histórico de vendas, entre outras. Mas não só as empresas de negócios ganharam com este avanço: a área da saúde saiu também bastante beneficiada, dado que é uma das áreas que mais dados angaria ao longo do tempo devido ao grande fluxo de informação que passa pelos hospitais, clínicas, etc.

Todos estes dados, que inicialmente pode parecer que apenas servem para consulta, se analisados de forma correcta, podem vir a fornecer um grande grau de conhecimento, que de outra forma estaria camuflado, o que pode ajudar bastante na tomada de decisões. Mas a crescente quantidade de informação que vai sendo armazenada ao longo do tempo excede claramente a capacidade humana de análise e compreensão, ou seja, torna-se impossível o uso de métodos de análise mais simples, ditos "tradicionais", como folhas de cálculo, assim como consultas definidas pelo utilizador [Tam, 1998].

Para colmatar esta falha surge o conceito de BI, já referido anteriormente. O intuito de um sistema contido nesta temática é de fornecer a capacidade de angariar os dados com um único propósito: o de ajudar no processo de tomada de decisões. As ferramentas de *OLAP (Online Analytical Processing)* constituem uma das principais formas de análise dos dados obtidos, mas estas são apenas um meio para atingir um fim, pois o conceito de BI é bastante mais abrangente.

Para se chegar a uma solução final, que possibilite ao utilizador uma correcta análise dos dados, não é suficiente uma simples ferramenta independente, mas uma combinação de conhecimentos e a implementação de diversas técnicas. Estas são, na sua maioria,

[1] Conceito por vezes traduzido para Inteligência Competitiva ou Inteligência Empresarial

referentes à constituição e gestão de bases de dados.

1.2 Motivação, objectivos

A área de BI é uma das áreas das tecnologias de informação que mais tem crescido nos últimos anos. A evolução das técnicas utilizadas e a crescente actualização e adição de novas funcionalidades a estas ferramentas levou a que as organizações percebessem o seu potencial e, por sua vez, a apostar neste tipo de soluções.

Os bons resultados obtidos por estas ferramentas na área dos negócios levou à expansão para outras áreas, como é o caso da saúde, em torno da qual a presente dissertação se vai desenvolver e onde a angariação de dados e de informação é também bastante considerável. Como tal, uma maior rapidez e facilidade na análise dos acontecimentos dentro de um hospital, centro de saúde, etc. pode levar a uma melhoria bastante grande nos cuidados prestados e na alteração de procedimentos.

Tendo em conta a necessidade de fornecer aos analistas uma plataforma de análise intuitiva e consistente, o principal objectivo deste trabalho é o desenvolvimento de uma ferramenta de OLAP que permita a observação de dados relativos a registos de enfermagem, mais concretamente na sua valência pediátrica.
Como tal, será necessário a implementação de um *Data Warehouse* onde os dados serão armazenados, tornando-se necessário a execução do processo de ETL(*Extraction, Transformation and Load*).
Por último, o analista deverá poder efectuar consultas aos dados, sem que seja obrigatório possuir qualquer conhecimento de bases de dados nem de código SQL. Estas consultas serão efectuadas por meio de tabelas que permitam a execução de operações OLAP, de forma a ser possível percorrer os dados através de diversos contextos.

1.3 Estrutura da dissertação

A estrutura desta dissertação consiste, numa primeira instância, em descrever os aspectos teóricos relativamente aos diversos componentes que permitem o desenvolvimento de um sistema de OLAP, sendo o capítulo 2 dedicado a esta descrição. No capítulo 3 é descrito o processo de desenvolvimento dos diversos componentes que constituem o modelo proposto. O capítulo 4 contém a avaliação do modelo, tanto ao nível de eficácia como de performance, onde são explicados os testes efectuados, indicados os resultados obtidos e

apresentados alguns melhoramentos possíveis. Por último, são feitas as considerações finais no capítulo 5. Neste, está incluída uma apreciação dos objectivos alcançados, assim como alguns aspectos a ter em conta, numa perspectiva de trabalho futuro.

Capítulo 2

Estado Da Arte

O desenvolvimento a que assistimos nos dias que correm leva-nos a ter que tomar decisões cada vez mais sustentadas e correctas. Tal encaminha-nos para uma procura incessante de mais e melhor informação.

O uso de bases de dados e a constante angariação de dados subjacente para salvaguardar e optimizar o funcionamento das organizações de diferentes áreas vieram trazer um grande avanço na perspectiva da colecta de dados que posteriormente se tornam numa grande fonte de conhecimentos.

O problema que se põe é o facto de nem sempre a informação que procuramos estar disponível de forma clara. Isto leva-nos a procurar soluções sofisticadas de forma a conseguir descobrir essa informação, contida no meio de uma quantidade infindável de dados armazenados.

Os *Data Warehouses (DW)*, aliados a aplicações *Online Analytical Processing (OLAP)*, estão contidos no rol de soluções para a resolução deste problema e são uma grande mais valia para os analistas, que vêem o seu trabalho bastante facilitado.

2.1 Data Warehouse

Ao longo dos anos, têm surgido diversas formas de definir um DW. Em termos teóricos e de acordo com W. H. Inmon, o seu principal arquitecto, *"a data warehouse is a **subject-oriented**, **integrated**, **time-variant**, and **nonvolatile** collection of data in support of management's decision making process"* [Inmon, 1992]. Por meio destes quatro termos é possível descrever as características de um DW, isto é:

- **Subject-oriented:** um DW é desenvolvido e organizado de modo a satisfazer as necessidades de análise de uma organização relativamente a um ou mais aspectos chave. Em relação ao trabalho realizado e descrito no capítulo 3, um dos aspectos chave é a taxa de prevalência relativa aos diversos diagnósticos, presentes nos registos de enfermagem.
- **Integrated:** um DW é, por norma, desenvolvido utilizando diversas fontes de dados externas, como bases de dados relacionais, folhas de cálculo, entre outras. Como tal, alguns problemas de consistência dos dados necessitam de resolução.
- **Nonvolatile:** um DW, por ser um sistema completamente distinto e separado do ambiente operacional, apenas permite o carregamento dos dados e a leitura dos mesmos. Operações de modificação e de remoção de dados não são permitidas.
- **Time-variant:** os dados são armazenados de modo a fornecerem uma perspectiva histórica dos dados.

Um DW é também considerado como uma base de dados relacional cuja estrutura obedece a determinadas especificações técnicas, explicadas mais à frente na secção 2.1.2, desenvolvidas exclusivamente para o suporte de ferramentas de análise e de apoio à decisão [Luján-Mora, 2005].

Mas, antes de mais, surge-nos uma pergunta sobre qual a necessidade do uso de DW em detrimento das bases de dados operacionais já implementadas e em funcionamento nas organizações, onde se pode aceder directamente aos dados. A resposta é relativamente simples: por questões de funcionalidade e performance. Isto é, as bases de dados operacionais estão normalmente associadas a sistemas de *Online Transactional Processing (OLTP)*, ou seja, são bases de dados desenhadas para gerir um grande número de transacções (`inserções`, `actualizações` e `remoções`) e um constante fluxo de dados com a preocupação de manter a consistência e recuperação dos dados. Como tal, são usadas

em operações do dia a dia das organizações, logo, estão em constante alteração, o que deteriora a precisão dos dados para o propósito de análise. Dado que as consultas de uma aplicação OLAP são bastante mais complexas que as de um sistema OLTP, executá-las sobre uma base de dados deste tipo iria causar uma quebra na performance e, por sua vez, diminuir substancialmente a capacidade desta em conseguir realizar as suas tarefas ditas 'operacionais' [Han and Kamber, 2000].

Em suma, os sistemas de apoio à decisão necessitam de ter acesso a dados históricos, requerem consolidação dos dados, ou seja, tem que haver agregação e resumo dos dados, normalmente provenientes de diversas fontes, de forma "limpa"e integrada. Tudo isto são características que os sistemas operacionais não conseguem satisfazer e daí não ser adequado o seu uso para a realização de tarefas de apoio à decisão.

2.1.1 Modelo Multidimensional

A multidimensionalidade está na base do bom funcionamento das ferramentas de apoio à decisão. Este modelo é também comummente designado por *Cubo de Dados* (**Data Cube**). Facilmente se percebe o porquê desta analogia entre o modelo multidimensional e um cubo, uma vez que nada melhor que esta figura geométrica para ilustrar o uso de múltiplas dimensões.

Importa referir que embora a representação do cubo ajude bastante na compreensão deste modelo, esta não deixa de ser uma ilustração um pouco simplista, dado que pela representação pode parecer que o cubo de dados está confinado a apenas três dimensões e na verdade eles podem ter n dimensões, tantas quantas as que forem necessárias para o desenvolvimento de um determinado problema.

Estas **dimensões** são perspectivas ou atributos, como mostra a figura 2.1, sobre os quais uma organização pretende manter registos [Han and Kamber, 2000]. Isto é, se quisermos manter registos, por exemplo, das vendas de uma determinada loja, podemos querer mantê-los relativamente às dimensões tempo, localização e tipo de produto, o que posteriormente vai permitir observar e filtrar essas mesmas vendas por mês ou por local de venda, entre muitas outras combinações.

Por norma, as dimensões são hierárquicas, ou seja, possuem **hierarquias** que são estruturas que definem vários níveis de granularidade dentro de uma dimensão e a relação entre estes mesmos níveis [Malinowski and Zimányi, 2006]. A granularidade pode ser maior

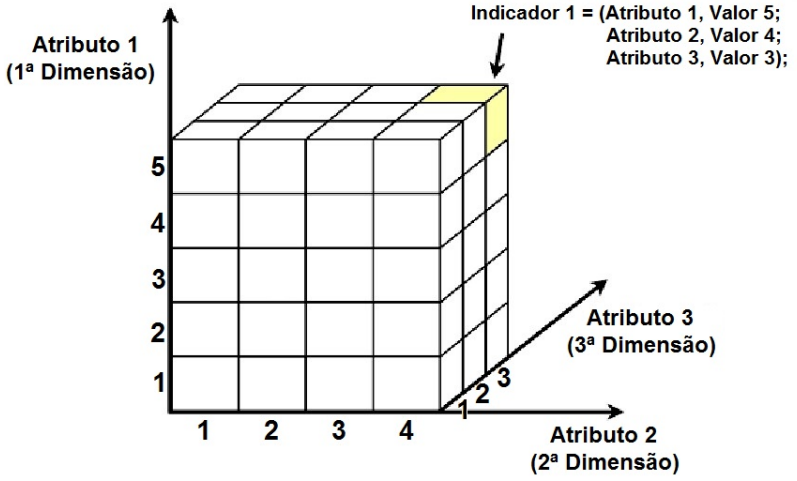


Figura 2.1: Representação de um cubo com três dimensões

ou menor conforme se sobe ou desce[1], respectivamente, na hierarquia. Por exemplo: a dimensão `Tempo` pode ter `horas` como nível mais baixo da hierarquia [Kimball and Ross, 2002], sendo por sua vez o nível com granularidade mais fina, e, por sua vez, ter o nível `semestres` ou `anos` como níveis mais altos e, por conseguinte, granularidade mais ampla. Cada nível da hierarquia possui os seus **membros**, os quais ajudam na filtragem dos dados dentro de uma dimensão, quando é necessário encontrar uma determinada situação mais concreta. Como tal, podem ser considerados um subconjunto de uma dimensão. Se estivermos, por exemplo, na hierarquia da dimensão `Região` e um dos níveis desta for `países`, Portugal, Espanha, entre outros, são membros possíveis dessa hierarquia.

Por sua vez, existem as **medidas** que são os valores quantificáveis, os quais são usados para analisar as relações entre as dimensões [Han and Kamber, 2000]. Recorrendo ao exemplo anterior, uma das medidas pode corresponder ao número de unidades vendidas ou então ao valor das vendas em dinheiro. Estas medidas são também os dados mais importantes, pois são estas que as organizações querem analisar em função das diversas dimensões.

[1] O nível de granularidade é inversamente proporcional ao nível de detalhe, ou seja, quanto maior a granularidade menor é o detalhe e vice-versa

2.1.2 Bases de dados Multidimensionais

Como já foi mencionado antes, um DW é uma base de dados relacional, mas com uma estrutura especial, mais propriamente uma estrutura multidimensional. Depois da descrição anterior sobre o modelo multidimensional, há que traduzi-lo para a realidade das bases de dados relacionais, criando desta forma bases de dados multidimensionais.

As duas formas mais conhecidas para representar um modelo multidimensional são o esquema em estrela (***Star Schema***) e o esquema em floco de neve (***Snowflake Schema***), embora haja um terceiro designado por constelação de factos (***Fact Constellation***), mas este último resulta de uma expansão dos outros.

Esquema em Estrela (Star Schema)

O esquema em estrela é o principal esquema utilizado e, ao mesmo tempo, o mais comum [Bhole, 2010]. Contém uma tabela central normalmente designada por tabela de factos (***Fact Table***), ou seja, é a tabela que contém os factos que podem corresponder às medidas ou que possibilitam o seu cálculo, e por sua vez é a que possui maior quantidade de dados e sem redundância dos mesmos. À volta desta estão tabelas mais pequenas, ditas "assistentes", que representam as tabelas das dimensões, conceito anteriormente explicado.

O esquema em estrela representado na figura 2.2 mostra a tabela de factos, neste caso para o exemplo das vendas de uma organização. Esta possui sempre as chaves primárias das tabelas de cada dimensão (que normalmente são geradas pelo sistema para uma melhor eficiência), assim como as medidas desejadas que nesta situação correspondem, tal como o nome indica, ao valor das vendas em euros e à quantidade de unidades vendidas. Como já foi dito anteriormente, esta tabela não possui redundância, pois cada linha da tabela corresponde a uma situação diferente, ou seja, a conjugação dos códigos de cada dimensão com os valores das medidas são sempre diferentes.

As dimensões para este exemplo são, como mostra a figura, `Localização`, `Tempo`, `Loja` e `Produto`. Cada uma contém um conjunto de atributos que, no caso da tabela `Localização`, correspondem a: `codigo_localizacao`, `cidade`, `freguesia`, `concelho` e `país`. Estes atributos vão posteriormente criar as hierarquias da dimensão, embora neste tipo de esquema estas não se encontrem explícitas. Nestas tabelas a redundância é muito comum, quando se trata deste género de esquema. Por exemplo, vão haver várias fre-

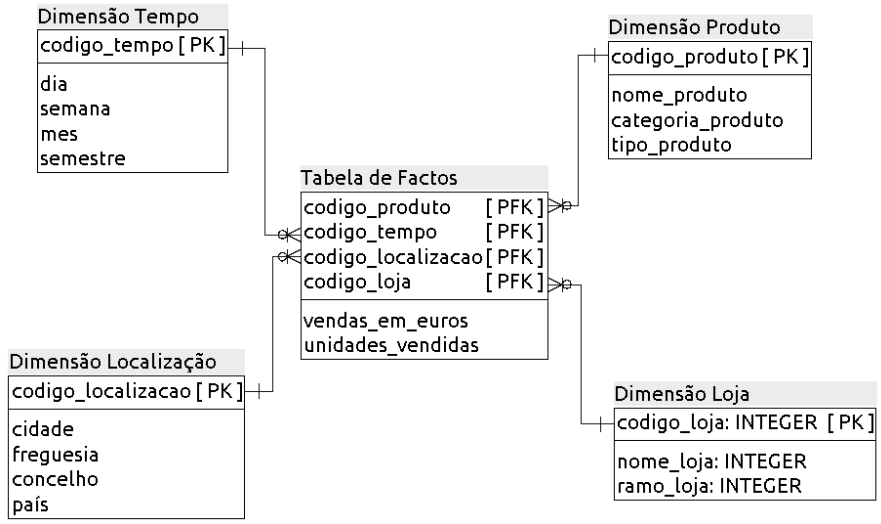


Figura 2.2: Exemplo de um esquema em estrela para a temática de vendas

guesias, do mesmo concelho e do mesmo país, o que vai provocar a redundância destes dois últimos atributos.

Esquema em Floco de Neve (Snowflake Schema)

A grande diferença deste esquema em relação ao anterior é a normalização das tabelas das dimensões: em vez de cada dimensão ser constituída por uma só tabela, estão divididas em várias tabelas.

Conforme podemos observar na figura 2.3, as dimensões `Localização` e `Produto` foram divididas em duas e três tabelas, respectivamente. Passou assim a existir uma tabela específica para o atributo `categoria` da dimensão `Produto` e passou a existir uma tabela para os atributos `freguesia` e `concelho`. Com esta normalização, algumas hierarquias passam a ser explícitas [Tam, 1998]. Esta vem trazer benefícios ao nível do espaço utilizado pelas tabelas das dimensões, sendo necessário menos espaço de armazenamento, diminuindo também a redundância destas tabelas. Contudo, dado que as tabelas estão separadas, são precisos mais *joins* aquando da execução de consultas, o que pode afectar bastante a performance. De referir ainda que a figura 2.3 é apenas um exemplo, pois mais divisões podem ser feitas além das que são mostradas, caso sejam necessárias.

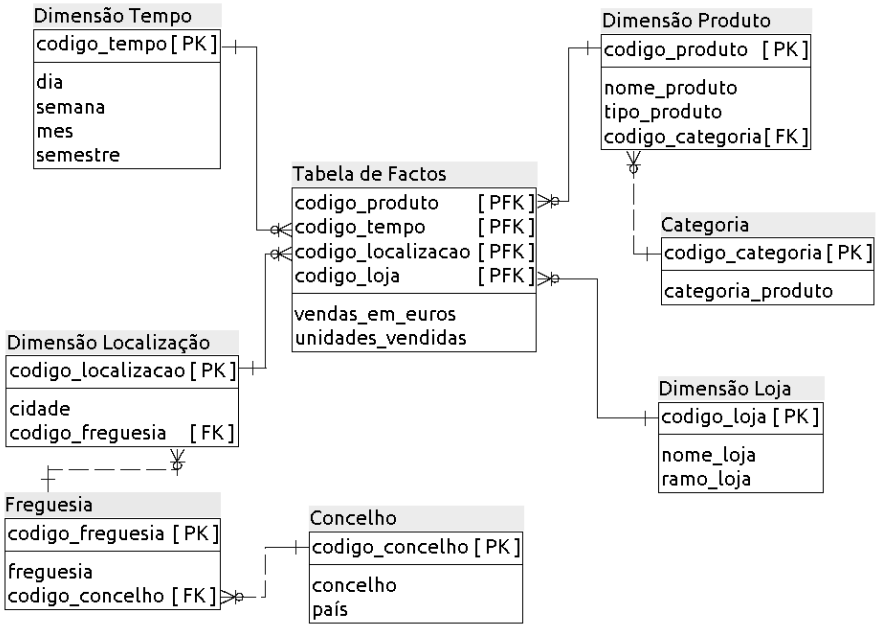


Figura 2.3: Exemplo de um esquema em floco de neve para a temática de vendas

Constelação de Factos (Fact Constellation)

Por último, existe ainda um esquema um pouco mais complexo, mas que não passa de uma extensão ou junção de dois ou mais esquemas em estrela. Este é designado por constelação de factos (***Fact Constellation***).

Tal como mostra a figura 2.4, foi adicionada uma segunda tabela de factos relativa aos envios de uma organização, onde esta partilha algumas das dimensões com a tabela de factos *vendas* anteriormente existente. Além das dimensões partilhadas (`Localização`, `Tempo`, `Produto`) podem existir dimensões exclusivas de cada tabela de factos, como é o caso das dimensões `transportadora` e `loja` que apenas dizem respeito à tabela de factos `envio` e `vendas` respectivamente.

2.1.3 Extracção, Transformação e Carregamento (ETL)

Posteriormente à estruturação do DW, inicia-se o processo de ETL *(Extracção, Transformação e Carregamento)*. Este processo é por norma o mais difícil e moroso quando se trata do desenvolvimento deste tipo de soluções [Reyes, 2010]. O principal objectivo é a

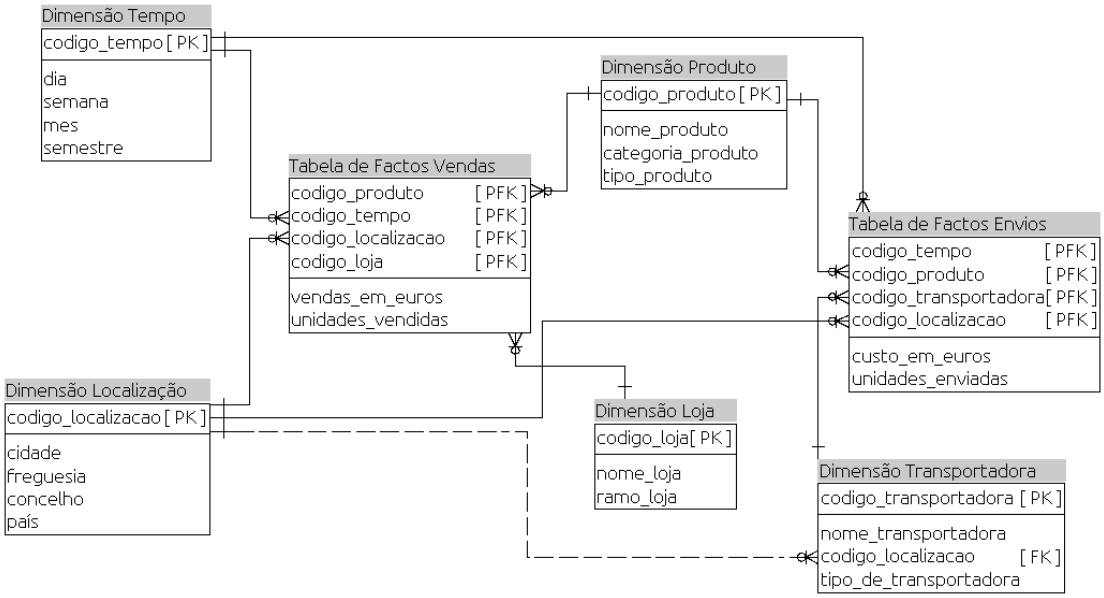


Figura 2.4: Exemplo de uma constelação de factos para a temática de vendas e envios

angariação e a transferência dos dados de diversas fontes de uma organização para o DW, ou seja, juntar os dados, por norma heterogéneos, para uma representação homogénea que permita processos de análise eficazes e eficientes.

Por vezes, antes da realização deste processo, é necessária a realização de um outro, designado por limpeza dos dados (***data cleaning***), isto é, deve ser realizada uma limpeza dos dados antes que estes sejam extraídos, transformados e finalmente carregados para o DW. Tal deve-se ao facto de os dados nem sempre se apresentarem de forma correcta ou completa, criando problemas de consistência.

Extracção

O primeiro passo é onde se identificam os dados que interessam de cada uma das fontes identificadas. Requer algumas precauções, devido ao facto de na prática ser necessário ter em atenção alguns aspectos, como não sobrecarregar as fontes, dado que podem estar a ocorrer outras actividades ao mesmo tempo, e ter cuidado para não interferir com as configurações destas [Vassiliadis and Simitsis, 2009]. Uma forma de prevenção, será executar este primeiro passo em alturas de pouca actividade, normalmente durante a noite.

A melhor forma de executar este passo é a criação de uma imagem dos dados necessários por parte da aplicação de ETL. Desta forma, não há qualquer risco para os dados que estão na fonte, pois não sofrem qualquer alteração.

Transformação

Este passo é responsável pela uniformização dos dados, na medida em que, como estes são provenientes de diversas fontes, é muito comum existir uma mesma denominação para objectos diferentes, ou então precisamente o contrário: denominações diferentes para o mesmo objecto. Neste passo podem também ocorrer problemas ao nível do tipo dos dados, como diferentes tipos de dados para a mesma informação, assim entre outras inconsistências que surgem quando se tenta juntar um nível considerável de informação dispersa num só local.

Uma variedade de funções são utilizadas de modo a padronizar todas as fontes antes do passo seguinte, como reformatação, junção de informação, normalização, optimização, entre outras [Vassiliadis and Simitsis, 2009].

Carregamento

Este último passo diz respeito à passagem dos dados, depois de transformados, para as suas respectivas tabelas no DW. Após o primeiro carregamento, os seguintes são por norma mais rápidos, pois apenas é feito um refrescamento dos dados através da comparação dos dados já existentes e dos dados provenientes das fontes, sendo apenas adicionados os novos dados. Desta forma é garantida uma base histórica muito importante, tornando o processo simultaneamente mais rápido e eficaz.

2.1.4 Arquitectura de um Data Warehouse

Na arquitectura de um DW podemos considerar a existência de quatro camadas. A primeira camada *"Data Sources"* diz respeito à angariação dos dados através da utilização de diversas fontes de dados. A segunda camada *"Data Storage"*, é a que contém a base de dados relacional com todas as características anteriormente descritas, que é gerida por sistemas de bases de dados relacionais [Tam, 1998]

Tal como mostra a figura 2.5, a segunda camada está representada na segunda fase da

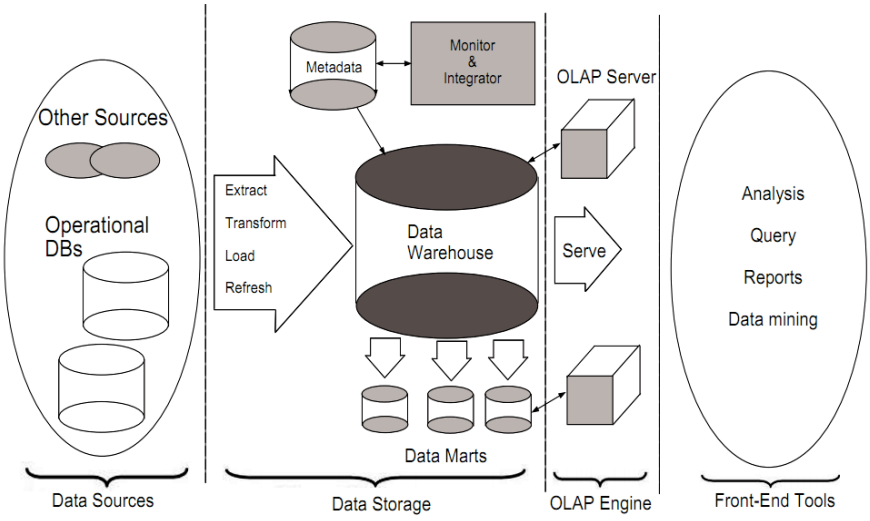


Figura 2.5: Arquitectura de um Data Warehouse [Taleb, 2011]

imagem onde também são mostrados diversos *data marts*. Estes podem existir ou não em caso de necessidade, pois cada um corresponde a um DW, mas mais pequeno, dizendo respeito a um departamento especifico da dentro de uma organização [Taleb, 2011].

A terceira camada *"OLAP Engine* é bastante importante e foi criada especialmente para este tipo de ferramentas de apoio à decisão [Chaudhuri and Dayal, 1997], ou seja, opera sobre dados em estruturas multidimensionais, permitindo a manipulação destas por múltiplos utilizadores [Moulton,] fornecendo-lhes transparência na navegação e criação de relatórios. Esta camada designa-se por *servidor OLAP* e existem diversas formas para a sua implementação, embora as mais usadas sejam o MOLAP e o ROLAP.

A principal diferença entre elas recai sobre a forma como os dados são armazenados, isto é, de forma relacional (***R**elational* ***OLAP***) ou multidimensional (***M**ultidimensional* ***OLAP***). Estes são os dois mais importantes, mas existe também um terceiro, cujo armazenamento é híbrido (***H**ybrid* ***OLAP***), ou seja, é uma mistura dos dois anteriores, tentando tirar partido das vantagens de cada um.

A última camada *"Front-End Tools* é constituída pelas ferramentas de análise onde são executadas as consultas por parte do cliente.(ver secção 2.2)

Relational OLAP

Este tipo de servidor fornece um *middleware* especializado de maneira a estender as capacidades dos servidores relacionais para que estes consigam suportar mais eficientemente as consultas multidimensionais características das ferramentas OLAP. Como tal, serve de intermediário entre o servidor relacional *back end*, onde estão armazenados os dados, e o cliente *front end*, ou seja, as ferramentas de análise. [Chaudhuri and Dayal, 1997]

Depois de carregados os dados para a base de dados, são criados índices de maneira a optimizar e reduzir o tempo de acesso necessário para a execução das consultas. Quando é submetida uma análise multidimensional por parte do cliente, o servidor OLAP transfere o pedido dinamicamente, através de linguagem SQL. Estes servidores têm que suportar várias nuances dentro da linguagem SQL, uma vez que cada sistema de gestão de bases de dados, onde está implementada a base de dados relacional, possui as suas especificações, o que por vezes pode trazer algumas incompatibilidades e problemas de performance. Posteriormente, o pedido é encaminhado para processamento pela base de dados relacional, onde o resultado é gerado na forma multidimensional e enviado novamente para o cliente [Maddi and Khan, 2007].

É fácil concluir que o armazenamento dos dados neste tipo de servidor OLAP é feito na própria base de dados relacional, daí o seu nome. Como exemplo de servidor baseado em ROLAP temos o servidor *open source* Mondrian da empresa Pentaho [Pentaho, d].

Multidimensional OLAP

O MOLAP baseia-se no armazenamento de vistas multidimensionais: em vez de funcionar como intermediário entre o DW e o cliente, como o anterior, os dados são guardados de forma multidimensional em estruturas do tipo *array*.

No funcionamento destes servidores, os dados são copiados do DW e são pré-calculadas e armazenadas nestas estruturas agregações de diversas combinações das dimensões existentes. Se por um lado esta tarefa pode ser algo demorada devido a todas as combinações que têm de ser efectuadas, por outro, com este tipo de abordagem, o tempo de resposta às consultas efectuadas do lado do cliente é bastante mais curto, dado que não é necessária nenhuma tarefa posterior aquando da sua execução [Westerlund, 2008].

Este tipo de servidores pode ser bastante eficaz, dado que é bem mais simples fazer uma procura numa estrutura do tipo *array* do que em tabelas [Tam, 1998]. Mas também pode

trazer bastantes problemas, principalmente ao nível de utilização de espaço, dado que muitas das combinações que são pré-calculadas podem não ter nenhum valor associado, fazendo com que na realidade um *array* bastante longo não possua informação útil, utilizando espaço desnecessariamente. Cada vez que se adiciona uma nova dimensão, o espaço utilizado aumenta significativamente, o que pode levar também a um aumento descontrolado do volume de dados. Por tudo isto, o MOLAP é, por norma, utilizado quando existem poucas dimensões.

Como exemplo deste tipo de servidores MOLAP, temos o Palo da empresa Jedox [Jedox,] que, à semelhança do anterior, também é *open source*.

ROLAP vs. MOLAP

Optar por um destes dois sistemas é algo difícil, já que ambos apresentam vantagens e desvantagens. Por norma, os servidores ROLAP tendem a ser mais utilizados, devido ao facto de a diferença de performance entre eles, quando existe um número reduzido de dimensões, não ser significativa. Além disso, se houver um aumento no volume dos dados ou no número de dimensões, a sua escalabilidade é também um ponto a favor deste tipo de servidores. Isto deve-se ao facto de não ser necessário o cálculo prévio das agregações, característico dos servidores MOLAP, respondendo de forma mais eficiente ao aumento significativo do volume de dados [Westerlund, 2008].

2.2 Operações OLAP

As ferramentas de OLAP permitem ao utilizador comum a possibilidade de navegar pelos dados alojados num DW. Como tal, necessitam de possuir algumas funcionalidades base, além de outras que podem ser disponibilizadas conforme a ferramenta utilizada.

Com os dados organizados de forma multidimensional e existindo, como já foi referido em 2.1.1, dimensões hierarquicamente organizadas, a flexibilidade de navegação por entre os dados é bastante grande. Para tal, existem algumas funcionalidades base características destas ferramentas, como: ***drill-up***, ***drill-down***, ***slice***, ***dice*** e o ***pivot***.

2.2.1 Drill-up ou Roll-up

O *drill-up* ou *roll-up* é a possibilidade de subir na hierarquia de uma dimensão em particular ou mesmo de remover uma dimensão. Como mostra a figura 2.6, passamos de uma vista por `cidades`, no cubo à esquerda, para uma vista por `distritos`, no cubo da direita, ou seja, todas as cidades foram agrupadas para os respectivos distritos, resultando numa vista dos dados "por distritos" em vez de "por cidades". Por outro lado, também seria considerado *drill-up* se a dimensão `Local` fosse removida por completo, o que agruparia os dados apenas por `Tempo`, que é a outra dimensão disponível neste caso.

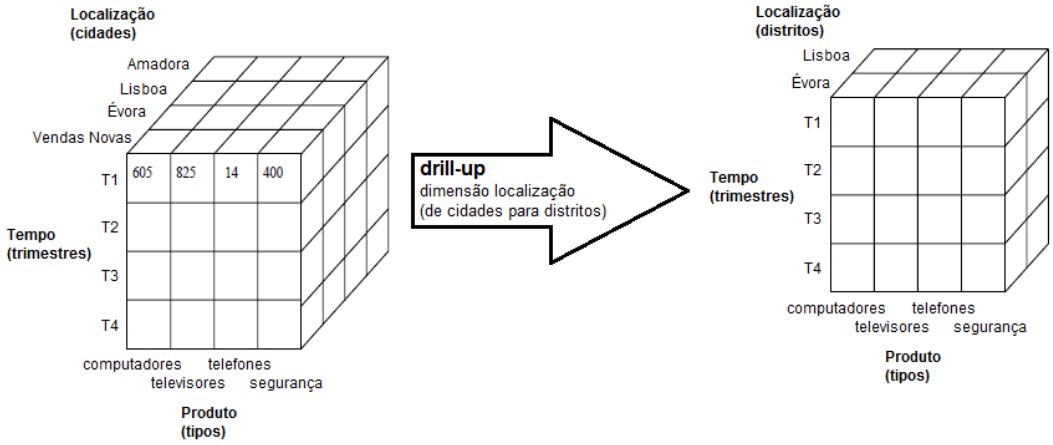


Figura 2.6: Operação OLAP Drill-up

2.2.2 Drill-down

A operação *drill-down* é exactamente o contrário da anterior: é a possibilidade de aumentar o nível de detalhe dos dados, descendo na hierarquia de uma dimensão ou acrescentando uma nova dimensão às já existentes na vista actual. A figura 2.7 ilustra bem esta situação, na qual deixamos de ter uma vista por trimestres para passarmos a ter uma vista mais detalhada, por meses.

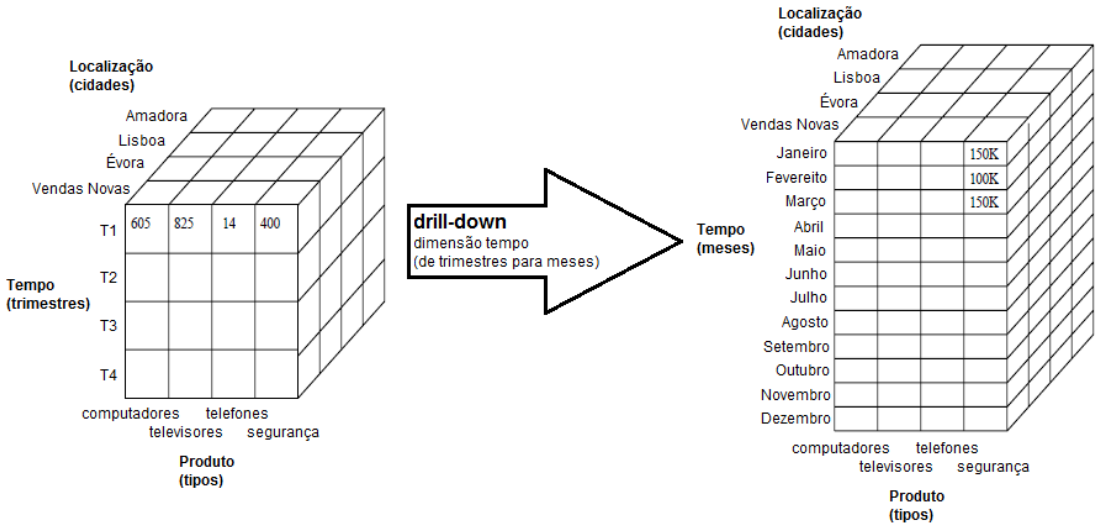


Figura 2.7: Operação OLAP Drill-down

2.2.3 Slice e Dice

As operações *slice* e *dice* são duas funcionalidades bastante parecidas. Ambas permitem efectuar "cortes" na visualização dos dados, ou seja, quando efectuamos um *slice* estamos a querer visualizar os dados relativos a um ou mais membros específicos de uma dimensão. Como podemos observar na figura 2.8, foi escolhido uma vista dos dados apenas relativos ao segundo trimestre. Relativamente ao *dice* é basicamente o mesmo processo, mas referente a um ou mais membros específicos de duas ou mais dimensões diferentes, criando uma espécie de subcubo. No caso da figura 2.8 podemos observar que na dimensão `Local` foram escolhidos os membros `Évora` e `Lisboa`, na dimensão `Tempo` os membros `T1` e `T2` e, na dimensão `Produto`, os membros `Computador` e `Televisor`.

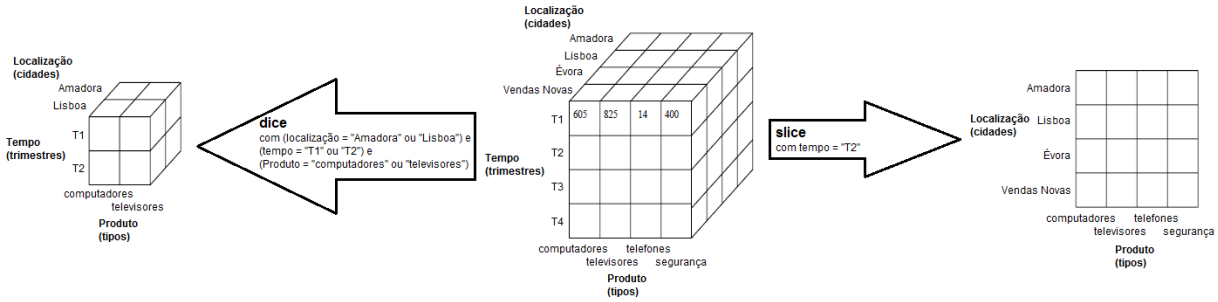


Figura 2.8: Operações OLAP Slice e Dice

2.2.4 Pivot ou Rotação

Esta funcionalidade, como o nome indica, permite a troca dos eixos de um cubo criando uma vista alternativa dos mesmos dados, isto é, se estivermos a observar, como mostra a figura 2.9, por `Localização` e `Produto`, a vista alternativa destes mesmos dados seria observar por `Produto` e `Localização`. Esta funcionalidade é bastante útil para poupar tempo, dado que para efectuar esta mesma operação seria necessário remover toda a vista actual e voltar a seleccionar a vista alternativa.

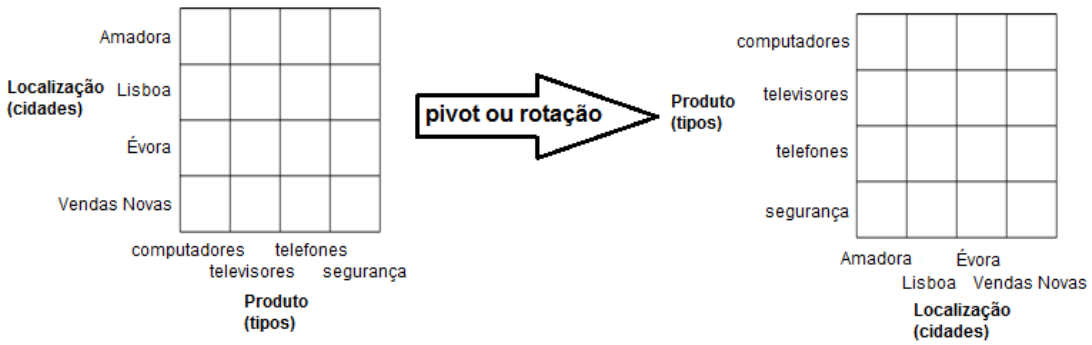


Figura 2.9: Operações OLAP Pivot ou Rotação

2.2.5 Outras Operações

As operações anteriormente descritas são as mais comuns e praticamente essenciais numa ferramenta de OLAP. Mas existem mais operações disponíveis para o utilizador como:

- O **Drill-through** que permite uma vista sobre a fonte dos dados, isto é, através desta funcionalidade é possível analisar a fonte dos dados que está por trás de uma determinada agregação [Gunderloy and Sneath, 2001].
- O **Drill-across** que permite executar consultas envolvendo mais do que uma tabela de factos.
- **Rankings** que permitem ordenar os dados que estão a ser analisados por ordem crescente ou decrescente.

No próximo capítulo, algumas destas operações serão melhor ilustradas através de exemplos práticos resultantes da análise do modelo proposto para esta tese.

2.3 Trabalho Relacionado

A adaptação deste tipo de ferramentas para a área da saúde não é propriamente nova e tem vindo a ser equacionada e estudada. Existem algumas diferenças relativamente à área dos negócios, onde são normalmente utilizadas. Desde logo, um dos problemas principais inerentes ao uso de informação médica em ferramentas de análise é garantir a privacidade dos pacientes, uma vez que se trata de informação bastante sensível [Puhr, 2002]. Outro problema reside nos valores quantificáveis para análise. Na área dos negócios é relativamente fácil encontrar valores quantificáveis que vale a pena analisar. Por exemplo, as quantidades de vendas efectuadas, tanto em termos de unidades vendidas como de valores monetários. No que respeita à área da saúde são necessárias pesquisas para se conseguir encontrar valores quantificáveis que tenham algum significado e cuja análise ajude no apoio à tomada de decisões. Em suma, um DW para o âmbito da saúde em termos funcionais e tecnológicos não é muito diferente de um DW virado para outras áreas, sendo, no entanto, mais complexa a obtenção e a posterior interpretação dos dados [Puhr, 2002].

Em *"The Clinical Data Warehouse"* [Puhr, 2002] foi proposto um modelo de dados multidimensional para análise de dados angariados no departamento de Cirurgia do

Hospital Geral de Viena.

Através do modelo relacional, já existente para guardar os relatórios de cirurgia, foram escolhidas as tabelas que realmente interessavam para o desenvolvimento do projecto, mais precisamente os dados referentes ao cirurgião, ao tempo e aos diagnósticos e terapias. Posteriormente foi aplicado o processo de ETL já descrito em 2.1.3 para optimização dos dados para a sua representação multidimensional.

Através dos dados disponíveis nas tabelas retiradas do modelo relacional foram identificadas cinco dimensões:

- Tempo;
- Cirurgião;
- Departamento;
- Sala de Operação;
- Grupo de Operação;

Foram também identificados os factos que neste caso de estudo correspondem a intervalos de tempo:

- Duração da anestesia;
- Período total dentro da sala de operações;
- Período dentro da sala de operações antes do começo da cirurgia;
- Período dentro da sala de operações depois do fim da cirurgia;
- Duração da cirurgia;

De seguida são propostos dois modelos multidimensionais: um, baseado num esquema em floco de neve (referido na secção 2.1.2), embora só a dimensão do tempo esteja normalizada; outro, baseado num esquema em estrela (referido na secção 2.1.2). Dada a simplicidade do modelo, nenhum se sobrepôs ao outro em termos de eficiência, pelo que não é referido qual dos dois foi utilizado na solução final.

Outro exemplo disponível da implementação de um modelo multidimensional no âmbito da saúde, é o descrito em *"A framework for designing a healthcare outcome data warehouse"* [Parmanto et al., 2005]. Neste artigo é proposto um modelo multidimensional para o desenvolvimento de um DW para analisar os valores alcançados pelas terapias de reabilitação efectuadas sobre os pacientes, provenientes do Centro do Serviço de Reabilitação do Centro Médico da Universidade de São Petersburgo.

O esquema proposto é mesmo referido como sendo um possível modelo para outros DW na área da saúde. Este modelo é baseado num esquema em estrela com nove dimensões, entre as quais estão as dimensões Género, Idade, Diagnósticos, Data. Como medidas propostas, estão o número de visitas ao serviço por parte do paciente e os valores obtidos por um questionário sobre a qualidade de vida dos pacientes em termos de saúde, designado por SF-36 .

2.4 Resumo

Neste capítulo foi feita uma análise teórica sobre a temática de OLAP e que bases são necessárias para o desenvolvimento de uma ferramenta de apoio à decisão onde se englobam estas aplicações.

O conhecimento sobre DW torna-se imperativo, pois é em torno destes que estas ferramentas se focam. Como tal, é descrito o modelo multidimensional característico, pois é um modelo que agiliza o processo de consultas que são efectuadas na análise dos dados, e são também descritas e comparadas as duas arquitecturas mais importantes.

Nas últimas duas secções deste capítulo são relatadas e ilustradas as principais operações características das aplicações OLAP e é efectuado um ponto de situação relativamente ao que já existe em termos de trabalho relacionado na área da saúde, pois o modelo proposto no capítulo seguinte insere-se no âmbito hospitalar.

Capítulo 3

Modelo Proposto

As ferramentas de apoio à decisão estão por norma associadas à gestão, para ajudarem na dinamização e de alguma forma contribuírem para um aumento da produtividade das organizações, e, por esta via, aumentarem os seus lucros.

O modelo proposto nesta tese visa a construção e adaptação de uma destas ferramentas, neste caso, uma aplicação OLAP para análise multidimensional de dados relativos à área da saúde, mais propriamente provenientes de registos de enfermagem contidos numa base de dados relacional.

3.1 Análise do Problema

Antes de haver qualquer tipo de desenvolvimento foi necessário proceder ao estudo do problema, mais propriamente, saber o que se pretendia analisar e de que forma era importante fazê-lo. Nesse sentido, o projecto baseou-se na análise de três medidas, isto é, três valores quantificáveis que correspondem a:

- **Taxa de Prevalência (TP)** - saber a percentagem de pacientes que tiveram um determinado diagnóstico, resolvido ou não resolvido, no total de doentes com esse mesmo diagnóstico, por semestre.

- **Modificação Positiva no Estádio do Diagnóstico (MP)** - saber a percentagem de pacientes com um determinado diagnóstico resolvido, no total de doentes que tiveram esse mesmo diagnóstico, por ano.
- **Taxa de Efectividade na Prevenção (TEP)** - saber a percentagem de pacientes que deixaram de ter risco (ex. risco de cair) no total de doentes que estavam com esse risco, por ano.

Depois de definidas as medidas restava saber quais as dimensões, isto é, os filtros pelos quais interessa avaliar os dados quantificáveis. Para este modelo as dimensões são as seguintes:

- Tempo - para efectuar a análise por ano e semestre;
- Geográfica - para efectuar a análise por freguesia;
- Diagnóstico - para efectuar análise por cada diagnóstico;
- Idade - para efectuar a análise por idades, mas com a particularidade de até aos 2 anos de idade ser efectuada por meses e, posteriormente até aos 18, por anos;
- Género - para efectuar a análise por sexo do paciente;

De salientar ainda, no âmbito da análise do problema, que todos os dados disponíveis se referiam apenas à unidade de pediatria, o que diminuiu o número de diagnósticos a ter em conta sendo que, para além disso, cada medida tinha o seu conjunto de diagnósticos específicos. Assim, foi necessário proceder à selecção dos diagnósticos para cada uma das medidas.

3.2 Ferramentas Utilizadas

Para a realização deste trabalho foram utilizadas diversas ferramentas, desempenhando cada uma a respectiva função conforme a fase do trabalho. À excepção das ferramentas utilizadas para o desenvolvimento da base de dados (Mysql e Power Architect), as restantes foram desenvolvidas e são propriedade da empresa Pentaho [Pentaho, e]. Ao longo de cada fase, será feita uma breve descrição de cada uma das ferramentas utilizadas. De referir ainda que todas estas ferramentas já atingiram um nível de maturidade

bastante elevado sendo possível a sua utilização sem prejudicar o correcto funcionamento do trabalho desenvolvido e para além disso estão também disponíveis para a comunidade e são *open source*. Por estas razões, foram escolhidas para serem utilizadas no desenvolvimento do corrente modelo proposto.

3.3 Método de desenvolvimento

Para o desenvolvimento deste modelo foi feito um estudo de alguns exemplos já existentes na área das vendas, que são os exemplos mais comuns, de modo a conseguir efectuar uma adaptação correcta e capaz de responder a esta nova área.

Como tal, o trabalho foi estruturado de forma incremental, passando por diversas fases:

- Construção do repositório multidimensional (secção 3.4);
- Processo de ETL (secção 3.5);
- Criação dos cubos (secção 3.6);
- Criação de vistas pré-definidas para análise dos dados (secção 3.7);

Das quatro fases que constituem o trabalho, a estruturação da base de dados multidimensional e a respectiva alimentação a partir da fonte são fundamentais para uma boa análise subsequente dos dados e, consequentemente, a parte mais importante do presente modelo e a mais exigente em termos de volume de trabalho.

3.4 Construção do repositório multidimensional

Com base na análise do problema, começou por se desenvolver o repositório multidimensional. Como já foi explicado na secção 2.1.1, um repositório multidimensional é uma base de dados que respeita as características do modelo multidimensional. Desta forma, para desenvolver a base de dados foi usado um dos sistemas de gestão de bases de dados (SGBD) mais conhecidos, designado por Mysql [MySQL,], por ser bem documentado e pela vantagem da sua utilização em situações anteriores.

A juntar ao SGBD, utilizou-se também a ferramenta Power Architect [SQLPOWER,] para a criação do esquema pretendido para a base de dados, pois esta forneceu uma

interface gráfica bastante intuitiva, que permitiu a criação das tabelas necessárias, dos atributos de cada tabela, das ligações entre elas (chave primária/chave estrangeira) e por fim a geração automática do código SQL, correspondente ao esquema elaborado, optimizando o processo de construção da base de dados.

3.4.1 Abordagem inicial

Numa abordagem inicial, apenas foi considerada uma das medidas (TP), sendo utilizado para implementação do modelo o esquema em estrela, como mostra a figura 3.1. Optou-se por esta abordagem, em detrimento de um esquema em floco de neve, devido ao facto da redundância dos dados não ser significativa, pois as tabelas das dimensões não continham um grande número de dados. Por outro lado, o número de *joins* necessários ao efectuar consultas num esquema em floco de neve podia trazer alguma perda de performance [Maddi and Khan, 2007], para além da maior simplicidade de construção que um esquema em estrela oferece relativamente ao floco de neve.

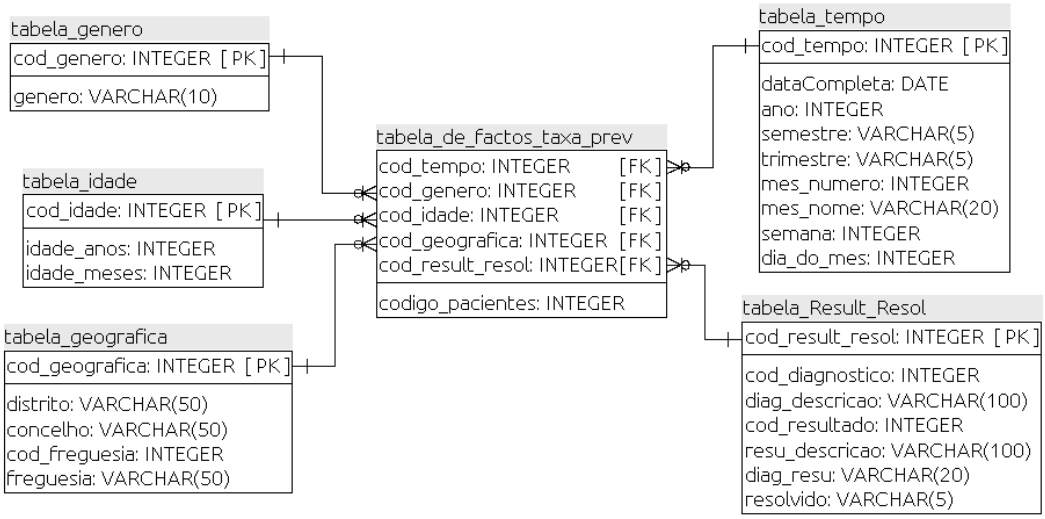


Figura 3.1: Esquema representativo da fase inicial da base de dados que constitui o DW para o modelo proposto

De referir que inicialmente a tabela de factos possuía, no lugar do campo `codigo_pacientes`, dois outros campos: um para a `taxa de prevalência`, outro para o número de

pacientes, por se considerar que estes dois indicadores poderiam ser logo calculados no processo de ETL. Quando o primeiro protótipo do sistema foi apresentado aos clientes do serviço, detectou-se este lapso na interpretação e na forma de cálculo dos dois indicadores, tendo sido pedida uma actualização em que o respectivo cálculo se efectuasse na terceira fase do trabalho, que irá ser explicada mais à frente.

3.4.2 Abordagem final

Posteriormente foram adicionadas mais duas tabelas de factos (`tabela_de_factos_modificacao` e `tabela_de_factos_efectividade`), uma para cada uma das medidas restantes (MP e TEP) e mais duas tabelas para as dimensões diagnóstico (`tabela_diagnostico_modificacao` e `tabela_diagnostico_efectividade`), exclusivas de cada medida, uma vez que cada medida possuía os seus diagnósticos respectivos, tal como foi explicado na secção 3.1. Com a inserção destas quatro tabelas o esquema evoluiu para uma constelação de factos, como representa a figura 3.2.

Para além de cada medida possuir os seus diagnósticos específicos, a medida `taxa de prevalência` tem a particularidade de possuir um maior número de dados, assim como um maior número de campos, como é possível constatar pela observação da tabela relativa à dimensão `Diagnóstico` (`tabela_Result_Resol`). Tal acontece de modo a permitir a criação de hierarquias para analisar a medida em questão, por diagnósticos e dentro de cada um, pelos casos resolvidos e não resolvidos. Esta propriedade não se verifica nas outras duas medidas, em que apenas interessavam os diagnósticos resolvidos.

Através da observação da figura 3.2, é possível também constatar a existência de uma tabela para cada dimensão referida na secção 3.1, ou seja, a `tabela_idade` para a dimensão `Idade`, `tabela_tempo` para a dimensão `Tempo`, `tabela_genero` para dimensão `Género` e `tabela_geografica` para a dimensão `Geográfica`. Todas estas são dimensões partilhadas, isto é, são comuns a todas as tabelas de factos, de maneira a permitir analisar as diversas medidas no mesmo contexto e pelos mesmos filtros. Apenas as tabelas dos diagnósticos são exclusivas de cada tabela de factos.

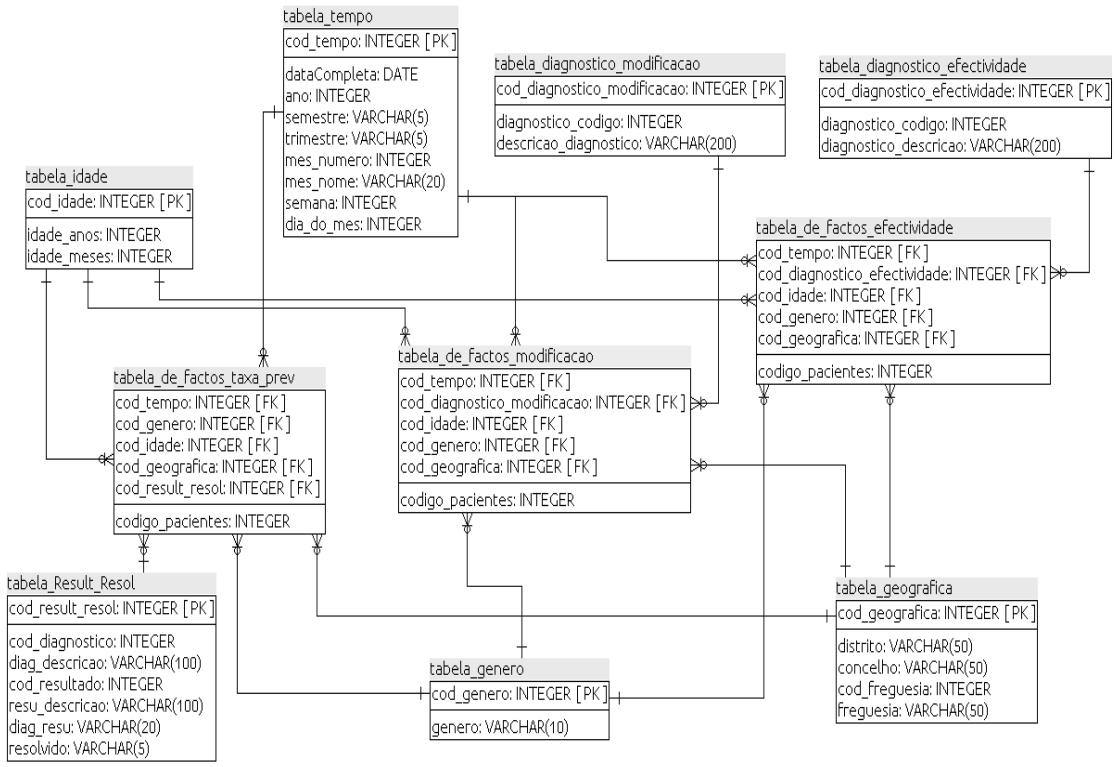


Figura 3.2: Esquema representativo da base de dados que constitui o DW para o modelo proposto

3.5 Processo de ETL

Esta fase do projecto, ao contrário do que normalmente acontece, não teve problemas de consistência dos dados e por conseguinte não foi necessário muito desenvolvimento para o processo de limpeza dos dados, uma vez que os dados usados foram provenientes de uma única fonte, uma base de dados relacional. Relativamente ao processo de ETL, verificou-se algo trabalhoso, sendo necessária a criação de diversas transformações e múltiplos testes, de modo a obter um correcto povoamento do repositório.

3.5.1 Análise da fonte de dados e ferramenta para ETL

Para realizar o processo de ETL foi necessário analisar e perceber quais as tabelas e os respectivos campos de cada uma da base de dados fonte, que continham os dados rele-

vantes e indispensáveis para povoar o DW. Como tal, foram identificadas e consultadas neste processo, catorze tabelas da base de dados de origem.

Depois de identificados os dados necessários passou-se ao processo de ETL propriamente dito. Para tal foi utilizada a ferramenta designada por *Pentaho Data Integration (PDI)* [Pentaho, c]. Esta permitiu criar uma "ponte" entre a fonte dos dados e o DW, através da criação de diversas transformações onde foram feitas as alterações necessárias aos dados, utilizando diversas operações, designadas por passos (***steps***). Por último, a execução destas transformações permitiu o carregamento dos dados para o DW.

Todas as ligações estabelecidas, tanto à base de dados fonte como ao DW, foram efectuadas recorrendo a drivers *JDBC (Java Database Connectivity)*, dado que se trata de uma ferramenta desenvolvida na linguagem *Java*.

3.5.2 Transformações referentes às tabelas das dimensões

Como podemos observar em qualquer um dos esquemas, representados nas figuras 3.1 e 3.2, as tabelas de factos são maioritariamente constituídas pelas chaves primárias das tabelas das dimensões, exceptuando os campos definidos para os factos. Dado que as dimensões fornecem os contextos para as tabelas de factos e por conseguinte para as medidas [Kimball and Caserta, 2004], o primeiro passo foi criar as transformações relativas às dimensões.

Para as dimensões das `Idades` e dos `Géneros` não foi necessário criar nenhuma transformação, pois como estas são estáticas e bastante curtas, os dados foram inseridos manualmente nas tabelas respectivas. No caso da tabela das `Idades`, tal como mostra a figura 3.2, a chave primária possui mais dois campos. O primeiro deles, `idade_meses`, foi povoado com valores entre 1 e 24, para os pacientes com a idade até dois anos, e o segundo campo, `idade_anos`, foi povoado com valores entre 3 e 18, para os restantes pacientes com idade superior a dois anos. Quanto à tabela `Géneros`, foi povoada com **F** para femininos e **M** para masculinos.

Dimensão Geográfica

Para a dimensão `Geográfica`, foi necessário criar uma transformação, como representado na figura 3.3. Apesar de se considerar estática, precisou de uma transformação específica, por ter um maior número de dados e por estar sujeita a mais alterações, relativamente

às anteriores.

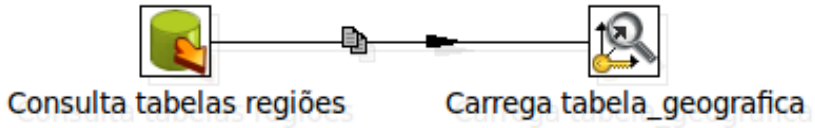


Figura 3.3: Transformação para tabela_geografica

No primeiro passo "consulta tabelas regiões", é feita, como o nome indica, a consulta à base de dados fonte, de modo a obter os nomes dos `Distritos`, dos diversos `Concelhos` referentes a cada distrito e das `Freguesias` referentes a cada concelho. Além dos respectivos nomes, são também obtidos os códigos de cada freguesia, os quais serão posteriormente usados nas transformações das tabelas de factos. Por fim é feito o carregamento dos dados obtidos para o DW através do passo (Carrega `tabela_geografica`), fazendo a correspondência das colunas obtidas da fonte com as colunas da `tabela_geografica` do DW. De referir ainda que, em carregamentos posteriores, este passo faz a comparação entre os dados provenientes da fonte e os já existentes na tabela da dimensão antes que estes sejam carregados. Desta forma apenas são adicionados os dados que foram alterados ou que ainda não existiam na tabela da dimensão, mantendo os restantes inalterados. Com este procedimento procura-se assegurar a eficiência do processo de carregamento.

Dimensão Tempo

A transformação da `tabela_tempo`, representada na figura 3.4, é diferente das restantes e um pouco mais complexa que a anterior. A principal diferença está no facto de não ser necessário qualquer acesso à base de dados fonte, tornando-a algo independente. Para povoar esta tabela foi definido um intervalo de tempo, baseado nas datas dos episódios de internamento, de acordo com uma metodologia que usou dois passos: no primeiro passo foi definida como data inicial o dia 1 de Janeiro de 2010; em segundo lugar foi definida uma transformação de forma a gerar uma sequência de dias que vão preencher a dimensão `Tempo`, cujo intervalo se procurou garantir que fosse suficientemente amplo para não requerer alterações frequentes.

O terceiro passo "Retira informação da data" efectua os cálculos necessários para retirar a informação requerida de cada data, correspondendo a:

- Dia;
- Número do Mês
- Ano;
- Semana do Ano;
- Trimestre;
- Semestre;

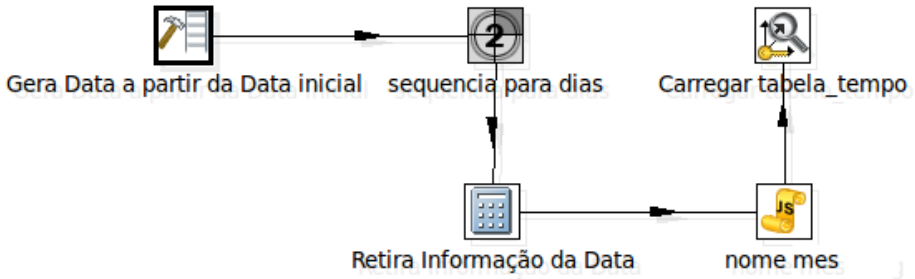


Figura 3.4: Transformação para tabela_tempo

Para facilitar a leitura, os números dos semestres e dos trimestres são precedidos de **S** e **T**, respectivamente. Esta nomenclatura é obtida no mesmo passo onde são efectuados os cálculos.

Finalmente, antes de ser efectuado o carregamento dos dados para a `tabela_tempo`, através do mesmo processo usado para a `tabela_geografica`, foi ainda usado o passo (nome mês em string) para obter os nomes dos meses através dos seus números.

Dimensão Diagnóstico

Para a dimensão `Diagnóstico` referente à `taxa de prevalência`, a transformação foi algo diferente e um pouco mais complexa, relativamente às restantes dimensões diagnósticos, devido às razões já explicadas em 3.4.2.

Tal como mostra a figura 3.5, inicialmente foram feitas duas consultas à base de dados fonte. Uma para obter os códigos dos diagnósticos e as respectivas descrições, referentes

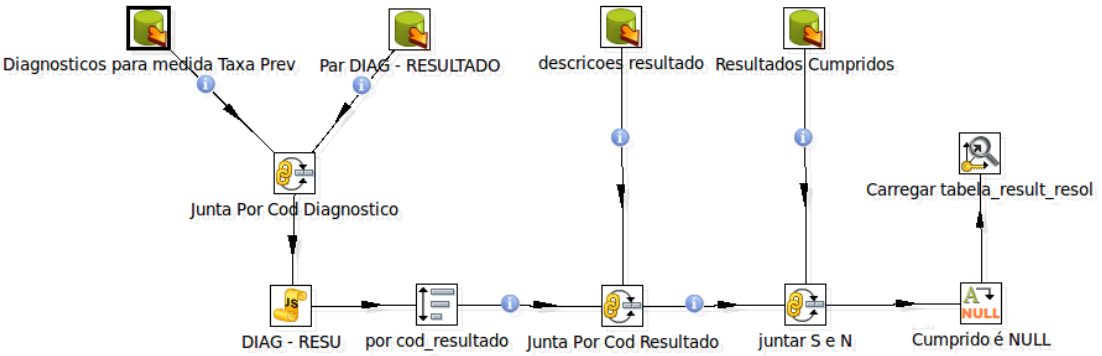


Figura 3.5: Transformação para tabela_result_resol

à medida em questão. Outra para obter os códigos dos diagnósticos e dos respectivos resultados, existentes nos planos de paciente, pois cada diagnóstico tem sempre um único resultado associado. Desta forma, apenas foram obtidos os diagnósticos realmente usados, não carregando dados desnecessários para a tabela desta dimensão.

De seguida foi feita a junção destas duas consultas através dos códigos dos diagnósticos, de modo a conseguir a correspondência entre os diagnósticos e os respectivos resultados e no passo seguinte foi criado o campo "Diagnóstico-Resultado". Posteriormente foram feitas mais duas consultas de modo a angariar toda a informação necessária para esta dimensão. A primeira junta a descrição dos resultados através da comparação entre códigos destes. A segunda é bastante importante, pois junta aos diagnósticos em questão: **S** ou **N**. Ou seja, é através desta junção que é adicionado o contexto de "resolvido" e "não resolvido" aos diagnósticos.

O último passo antes do carregamento também é importante, pois os diagnósticos seleccionados, apesar de terem sido diagnosticados a pacientes, não possuíam qualquer registo de resolução, o que fazia com que o campo "cumprido" ficasse sem valor (**NULL**). Como resolução para este problema, estes casos foram considerados como "não resolvidos",isto é, preenchidos com **N**.

De referir ainda a necessidade da criação de um passo intermédio (por `cod_resultado`), dado que aquando da utilização do passo para juntar duas tabelas, o PDI necessita, para evitar erros, que estas estejam ordenadas pela chave usada para a junção. Neste caso, através da comparação dos códigos dos resultados. No passo "descricoes resultado" não

é utilizado nenhum passo intermédio. É usada a cláusula *order by* para ordenar logo a consulta por este mesmo código.

3.5.3 Transformações referentes às tabelas de factos

Depois de efectuado o povoamento das dimensões, foi possível passar às tabelas de factos. Por norma, as transformações para este tipo de tabelas são bem mais complexas, como é possível constatar pela figura 3.6, dado que é necessário angariar toda a informação referente às dimensões, pois é através desta informação que é feita a comparação com o contexto das dimensões. Em suma, a informação é angariada da fonte, comparada com os dados contidos em cada dimensão e, através desta, constituído o contexto para cada facto, fornecendo a correspondência entre cada facto e as chaves primárias de cada dimensão.

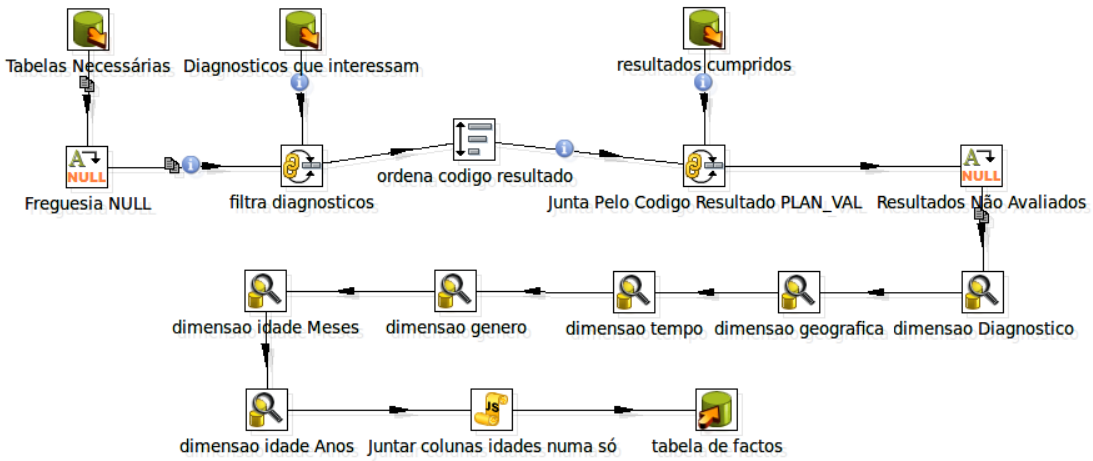


Figura 3.6: Transformação para tabela_de_factos_taxa_prev

Tabela de factos para taxa de prevalência

Para angariar toda a informação necessária, o primeiro passo desta transformação efectua a seguinte consulta:

```
SELECT PACIENTE.SEXO cod_sexo
```

```
      ,PACIENTE.POBLACION cod_freguesia
      ,EPISODIO.FECHAINI data_inicial
      ,Trunc ((EPISODIO.FECHAINI - PACIENTE.FECHANAC) / 365,0) as idade_anos
      ,Trunc (Months_Between(episodio.fechaini, paciente.fechanac)) as idade_meses
      ,ENF_PLAN_DIAG.DEFINICION cod_diagnostico
      ,ENF_PLAN_RES.DEFINICION cod_resultado
      ,ENF_PLAN_RES.CODIGO codigo_plan_res
      ,PACIENTE.CODIGO codigo_paciente
FROM PACIENTE
    ,EPISODIO
    ,ENF_PLANPACIENTE
    ,ENF_PLAN_DIAG
    ,ENF_DIAGNOSTICO
    ,SEXO
    ,ENF_PLAN_DIA_RES
    ,ENF_PLAN_RES
    ,ENF_RESULTADO_CTE
WHERE PACIENTE.CODIGO = EPISODIO.PACIENTE
  AND EPISODIO.CODIGO = ENF_PLANPACIENTE.EPISODIO
  AND ENF_PLANPACIENTE.PLAN = ENF_PLAN_DIAG.PLAN
  AND ENF_PLAN_DIAG.DEFINICION = ENF_DIAGNOSTICO.CODIGO
  AND PACIENTE.SEXO = SEXO.CODIGO
  AND ENF_PLAN_DIAG.CODIGO = ENF_PLAN_DIA_RES.DIAGNOSTICO
  AND ENF_PLAN_DIA_RES.RESULTADO = ENF_PLAN_RES.CODIGO
  AND ENF_PLAN_RES.DEFINICION = ENF_RESULTADO_CTE.CODIGO
  AND Trunc ((EPISODIO.FECHAINI - PACIENTE.FECHANAC) / 365,0) < 19
group by PACIENTE.CODIGO
        ,PACIENTE.SEXO
        ,PACIENTE.POBLACION
        ,EPISODIO.FECHAINI
        ,ENF_PLAN_DIAG.DEFINICION
        ,ENF_PLAN_RES.DEFINICION
        ,ENF_PLAN_RES.CODIGO
        ,Trunc ((EPISODIO.FECHAINI - PACIENTE.FECHANAC) / 365,0)
        ,Trunc (Months_Between(EPISODIO.FECHAINI, PACIENTE.FECHANAC))
order by ENF_PLAN_DIAG.DEFINICION
```

Como podemos facilmente constatar, através da interpretação do código SQL, com esta consulta a cada código de paciente (`cod_paciente`) corresponde:

- Um código sexo (`cod_sexo`);
- Um código de freguesia (`cod_freguesia`);
- A data do episódio (`data_inicial`);
- A idade do paciente quando foi feito o registo do episódio, em anos (`idade_anos`);
- A idade do paciente quando foi feito o registo do episódio, em meses (`idade_meses`);
- O código do diagnóstico (`cod_diagnostico`);
- O código do resultado (`cod_resultado`);
- O código com o qual se pode saber se o resultado foi cumprido ou não (`cod_plan_res`);

Na cláusula *Where*, além das junções das tabelas necessárias, através da correspondência entre as chaves primárias e chaves estrangeiras, acrescentou-se também uma verificação dos dados por idades, correspondente ao último *AND*, dado que só interessavam os casos relativos à pediatria. Daí a necessidade de extrair apenas os dados em que as idades fossem menores que 19 anos. Por último, a cláusula *group by* foi adicionada para prevenir os casos em que ocorresse o mesmo contexto, excluindo casos repetidos.

Através da execução da consulta anterior constatou-se que existiam bastantes pacientes sem código de freguesia associado. Como forma de resolver este problema, a todos estes casos foi atribuído o código da freguesia, existente na base de dados fonte, que corresponde ao valor "Privado".

De seguida procedeu-se à exclusão dos diagnósticos que não interessavam para a medida referente a esta tabela de factos. Esta opção foi tomada de maneira a diminuir o tamanho da tabela de factos, poupando espaço e ganhando performance no processo de análise. Para tal, foi efectuada uma consulta angariando os códigos dos diagnósticos referentes a esta medida, e executando, através do passo "filtra diagnosticos", a junção à direita (respeitando a orientação dos passos "Tabelas Necessárias" e "Diagnosticos que interessam" na figura 3.6), efectuando a comparação destes códigos, entre os dados obtidos pelas duas consultas, efectuadas nos passos "Tabelas Necessárias" e "Diagnosticos que interessam". Desta forma foi possível excluir todos os dados cujos códigos dos diagnósticos não fossem iguais aos contidos na segunda consulta.

Os seguintes quatro *passos* são referentes à junção da resolução ou não dos diagnósticos. Esta junção não foi efectuada logo com a primeira consulta devido ao facto de haver

muitos diagnósticos sem qualquer avaliação positiva ou negativa. Ou seja, se tivesse sido adicionada à cláusula *Where*, da primeira consulta, esta verificação, todos os casos anteriormente descritos tinham sido excluídos, o que não era do interesse do projecto, pois a interpretação requerida para estes casos era de considerar como não resolvidos. Como tal, foi feita a junção à esquerda, sendo os valores da esquerda provenientes do passo "ordena codigo resultado" e os da direita provenientes do passo "resultados cumpridos", entre o código `cod_plan_res` e o código `resultado` contido na tabela dos resultados cumpridos. Desta forma foi possível angariar os valores **S** e **N** para os diagnósticos resolvidos e não resolvidos, respectivamente. Para os campos que não possuíam qualquer avaliação, foram considerados como não resolvidos, sendo preenchidos com o valor **N** através do passo "Resultados Não Avaliados".

Posteriormente, é feita a comparação entre os dados angariados nesta transformação e os dados contidos nas tabelas das dimensões, obtendo-se de cada dimensão a chave primária correspondente para povoar a tabela de factos como chave estrangeira.

Antes do carregamento final, dado que a verificação da idade em meses era diferente da verificação da idade em anos, uma vez que são campos diferentes na tabela da dimensão `Idades`, foi necessário juntá-los numa só coluna através da seguinte verificação:

```
var cod_idade;

if(cod_idade_meses == 0)
   cod_idade = cod_idade_anos;
else
   cod_idade = cod_idade_meses;
```

Ou seja, se o código da idade em meses for igual a zero, é sinal que não foi encontrado o `cod_idade` (chave primária dimensão `Idades`) correspondente, o que quer dizer que o paciente em questão tem idade superior a 24 meses. Neste caso o código é o dos anos. Em caso contrário, devolve o código da idade em meses.

Para além das chaves primárias das diversas dimensões, são também carregados para a tabela de factos os códigos dos pacientes, correspondendo aos factos para esta tabela. Foi através destes que a medida `taxa de prevalência` foi calculada. Este cálculo pertence à fase seguinte e será explicado mais à frente.

Tabelas de factos para modificação e efectividade

As transformações para as duas medidas restantes são bastante semelhantes às anteriormente descritas. A grande diferença está no número de dados angariados na consulta inicial, pois como apenas interessavam os diagnósticos resolvidos, esta verificação foi logo efectuada na consulta inicial, não sendo necessário haver uma segunda junção como na transformação anterior.

Todo o restante processo é efectuado da mesma forma, à excepção do passo para a tabela referente à dimensão `Diagnóstico`, onde foi necessária uma alteração para as respectivas tabelas dos diagnósticos referentes às medidas em questão.

3.6 Criação dos cubos

Nesta fase procedeu-se à criação de um ficheiro XML, normalmente designado por *esquema*, responsável pela descrição dos cubos multidimensionais. Este ficheiro foi essencial para que o servidor OLAP utilizado, que neste caso se baseia na arquitectura ROLAP (ver secção 2.1.4), conseguisse interpretar os dados contidos no DW.

É com base no esquema criado que é feita a tradução das consultas efectuadas na ferramenta de análise, em linguagem *MDX (Multidimensional Expressions)*, especialmente criada para efectuar consultas a bases de dados OLAP [Pearson, 2002], para a linguagem SQL [Bouman and Dongen, 2009].

3.6.1 Ferramenta utilizada

Para desenvolver o esquema, foi utilizada a ferramenta designada por *Schema Workbench* [Pentaho, a], que fornece uma interface gráfica bastante intuitiva para a criação do ficheiro XML em questão. Esta conecta-se directamente ao DW, fazendo a ligação entre os vários componentes dos cubos e as respectivas tabelas.

Foi também neste esquema que ficaram definidas as hierarquias de cada dimensão, as medidas de cada cubo e a sua respectiva função de agregação, necessária para navegar dentro das hierarquias.

3.6.2 Esquema criado

O esquema definido para este projecto foi constituído por três cubos diferentes, um por cada tabela de factos.

Dimensões Partilhadas

Depois de criados os cubos foram criadas as quatro dimensões partilhadas entre os diversos cubos: as dimensões `Tempo`, `Geográfica`, `Género` e `Idades`.

A figura 3.7 é ilustrativa das hierarquias criadas para cada dimensão partilhada. Como podemos observar, para a dimensão `Tempo` foi criada uma hierarquia com os níveis representados na figura, sendo `Ano` o nível com menor detalhe e `Dia` como maior. Podemos também constatar que cada dimensão está associada à respectiva tabela. No caso da dimensão `Tempo` é feita a associação com a `tabela_tempo`. A mesma interpretação que é feita para a dimensão `Tempo` pode ser feita para as dimensões `Geográfica` e `Género`.

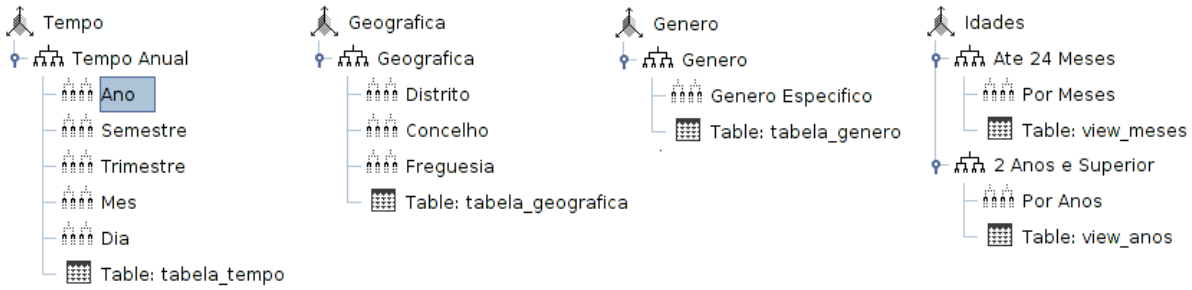


Figura 3.7: Dimensões partilhadas e respectivas hierarquias

A dimensão `Idades` tem duas particularidades relativamente às anteriores. A primeira é o facto de possuir duas hierarquias. Esta opção foi tomada de modo a permitir analisar os dados por meses, até aos dois anos, e por anos, para idades superiores a dois anos, de modo a cumprir a exigência proposta na secção 3.1. A segunda particularidade é o facto de estas hierarquias não estarem directamente associadas à `tabela_idades` e sim a duas `vistas`. Esta opção foi tomada, na estrutura desta tabela, quando a coluna `idade_meses` possuía valores, a coluna `idade_anos` ficava com valores **NULL**, e vice-versa. De maneira a prevenir que estes valores, sem significado, aparecessem na ferramenta de análise, foi criada uma vista para cada coluna de modo a permitir a exclusão destes.

Cada nível de uma hierarquia está associado a um campo da tabela em questão, tornando possíveis as funcionalidades de *drill*.

Cubos

Através da figura 3.8 podemos observar que o primeiro passo foi definir qual a tabela de factos associada ao cubo. Como se trata do cubo para a medida `taxa de prevalência`, este foi associado à respectiva tabela. Dado que as dimensões partilhadas foram definidas fora de qualquer cubo, para estas apenas foi definida a ligação entre os cubos e estas quatro dimensões, daí a diferença entre os símbolos. Como as dimensões relativas aos diagnósticos eram especificas de cada medida, foi criada a dimensão *Diagnostico-Resultado*, com a respectiva hierarquia.

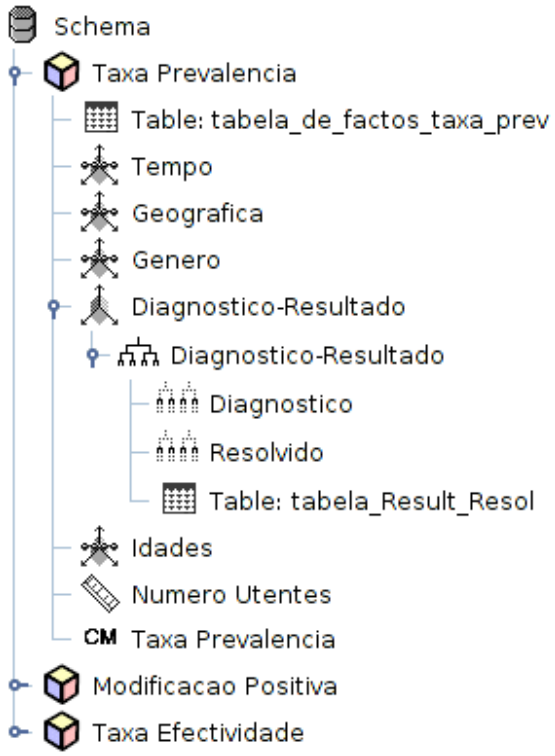


Figura 3.8: Estrutura de um cubo, respectivas dimensões e medidas

Por último, foi criada a medida *Numero Utentes* de modo a saber quantos utentes dizem respeito a um determinado contexto e foi através desta que foi possível o cálculo da taxa em questão. Esta medida é calculada através da contagem distinta (função de agregação

distinct-count) do número de `codigos_paciente`. Como a segunda medida é uma taxa, o seu cálculo foi mais complexo, sendo necessário criar um membro calculado (***calculated member***) com a seguinte fórmula em código MDX:

```
Case
 When [Diagnostico-Resultado].[Resolvido].CurrentMember.Level.Ordinal = 0
     Then 1
 Else ([Diagnostico-Resultado].[Resolvido].CurrentMember,
       [Measures].[Numero Utentes]) /
      ([Diagnostico-Resultado].[Resolvido].CurrentMember.Parent,
       [Measures].[Numero Utentes])
End
```

Ou seja, dado que esta taxa é calculada, dividindo o número de pacientes com um determinado diagnóstico pelo número total de pacientes e por fim multiplicando por cem, o código MDX acima representado permite o cálculo desta de forma dinâmica, através da medida `Numero Utentes`. Por exemplo, no caso de se querer analisar a `taxa de prevalência` para os casos em que um determinado diagnóstico tenha sido resolvido, o cálculo efectuado é a divisão do número de pacientes que estamos a observar (CurrentMember) pelo seu "pai" (CurrentMember.Parent), isto é, o número total de pacientes com esse diagnóstico e assim sucessivamente até chegarmos ao topo da hierarquia. Quando não houver mais níveis acima (CurrentMember.Level.Ordinal = 0) então o resultado devolvido é 1, pois como se trata de uma taxa, o valor nunca ultrapassa os 100%.

A estrutura dos restantes cubos foi similar, tendo sido apenas substituídas as tabelas de factos associadas, pelas respectivas tabelas, assim como as dimensões diagnósticos. Todo o restante processo foi igual ao já descrito.

3.7 Criação de consultas pré-definidas

Esta última fase do projecto diz respeito à ferramenta de análise propriamente dita. Optou-se pela utilização de um servidor OLAP que, como já foi referido anteriormente se baseou na arquitectura ROLAP. O servidor designa-se por *Pentaho BI Server* [Pentaho, b] tratando-se de uma aplicação Web, acedida através do browser. Aliado a este, foi utilizado como visualizador um projecto *open source* designado por *STPivot*

[StrateBI,]. Este possui uma interface gráfica bastante intuitiva, fornecendo ao utilizador todas as operações base que uma ferramenta de OLAP deve conter (ver secção 2.2), assim como a possibilidade de criar diversos tipos de gráficos, extrair as tabelas geradas para PDF ou para folha de cálculo, entre outras. A figura 3.9 mostra um exemplo desta mesma interface já com dados referentes ao modelo proposto neste projecto.

Por último, foram acrescentadas algumas consultas pré-definidas, de modo a facilitar o processo de análise e permitir alternar entre os diversos cubos, ou seja, entre as diversas medidas. A forma como podem ser acedidas está representada no lado direito da figura 3.9 e cada consulta diz respeito a um ficheiro do tipo `xaction`. Este é constituído por código XML e contém informação sobre o caminho para o ficheiro com o esquema, descrito na secção 3.6.2, o nome da base de dados referente ao DW e, por último, o código MDX, que contém a consulta desejada. No caso da figura 3.9 é representada a análise das duas medidas definidas (`Número de Utentes` e `Taxa de Prevalência`), no contexto de tempo e diagnósticos.

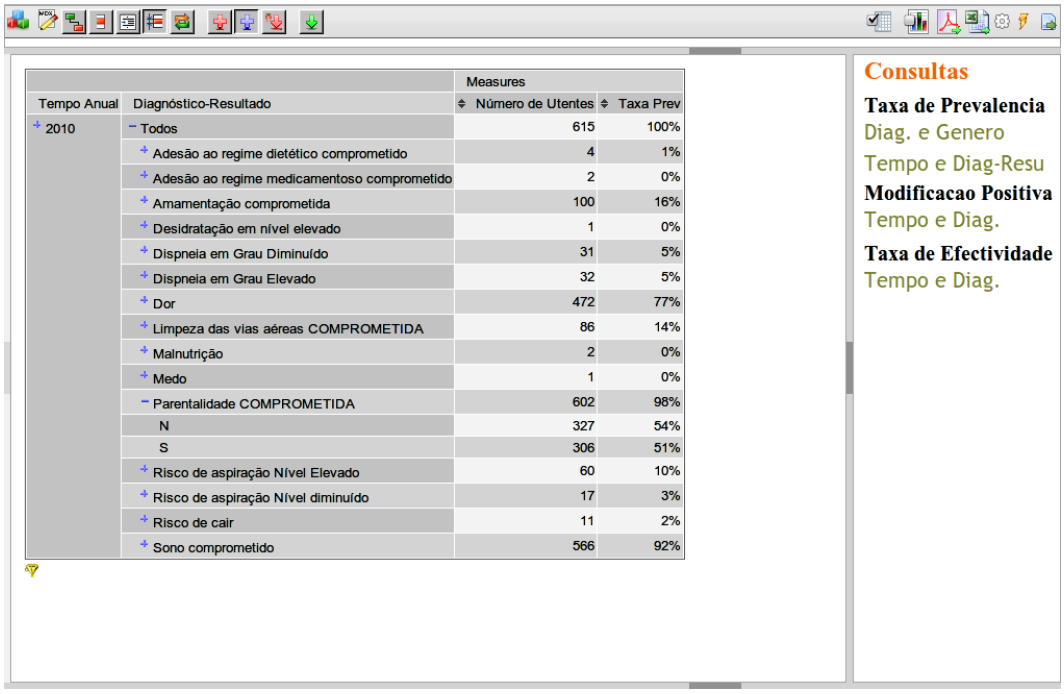


Figura 3.9: Consulta do número de utentes e taxa de prevalência pelas dimensões Tempo e Diagnóstico

A figura 3.10 ilustra uma outra consulta efectuada aos dados reais, embora esta seja

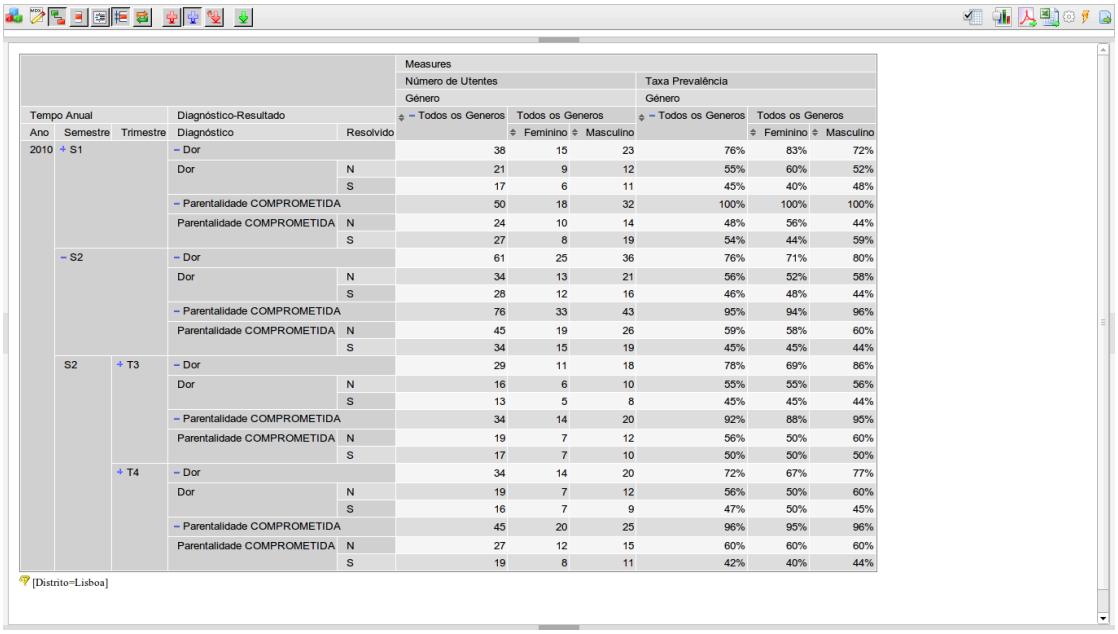
mais profunda que a representada na figura 3.9. Através desta, é possível analisar os valores referentes às medidas `Número de Utentes` e `Taxa de Prevalência`, por `Tempo`, `Diagnóstico`, `Género` e `Geográfica`, mas escolhendo algumas particularidades e efectuando algumas restrições à informação apresentada para alguns casos específicos. Analisando a tabela de dentro para fora, na dimensão `Diagnóstico-Resultado`, foi escolhido observar apenas os dados relativos a dois diagnósticos específicos: `Dor` e `Parentalidade COMPROMETIDA`. Além dos valores totais por diagnóstico, foi efectuado o *drill-down* até ao nível mais baixo da hierarquia desta dimensão, detalhando os valores por casos resolvidos e não resolvidos. Relativamente à dimensão `Tempo` o nível `Ano` da hierarquia foi ocultado, sendo os primeiros valores apresentados referentes ao segundo nível da hierarquia, os dois semestres do ano de 2010. Para além dos valores totais por semestre, são também mostrados os valores referentes aos trimestres do segundo semestre do ano 2010, através da execução de um *drill-down* no membro `S2` na hierarquia da dimensão `Tempo`. Quanto à dimensão `Geográfica`, como podemos observar na parte inferior da figura 3.10, foi efectuado um corte nesta mesma dimensão de modo a filtrar os dados apresentados. Isto é, todos os valores mostrados nesta tabela são apenas referentes ao distrito de Lisboa. Por último, a colocação da dimensão `Género` na parte superior da tabela, permite a observação das duas medidas, de uma forma geral e ao mesmo tempo por género específico, tornando a tabela mais concisa e fácil de analisar.

Como já foi referido em 3.7, o visualizador STPivot permite a criação de gráficos relativos às tabelas criadas. A figura 3.11 representa um dos tipos de gráficos possíveis, relativo à consulta efectuada na tabela apresentada na figura 3.10. Para este exemplo foi escolhido o gráfico do tipo circular, sendo possível observar cada linha da tabela através da sua representação gráfica. Cada gráfico possui uma descrição na parte inferior, mostrando o contexto em que se insere.

3.8 Resumo

Neste capítulo foi inicialmente efectuada a descrição do problema, assim como de todo o processo prático realizado até chegar à solução final. O processo foi dividido em quatro fases distintas, sendo feita uma descrição pormenorizada de cada uma.

A primeira fase diz respeito ao desenvolvimento da base de dados multidimensional que constitui o DW. Na segunda fase foram descritos os diversos passos do processo de ETL, necessário para a transformação dos dados provenientes da base de dados fonte



					Measures					
					Número de Utentes			Taxa Prevalência		
					Género			Género		
Tempo Anual			Diagnóstico-Resultado		– Todos os Generos	Todos os Generos		– Todos os Generos	Todos os Generos	
Ano	Semestre	Trimestre	Diagnóstico	Resolvido		Feminino	Masculino		Feminino	Masculino
2010	+ S1		– Dor		38	15	23	76%	83%	72%
			Dor	N	21	9	12	55%	60%	52%
				S	17	6	11	45%	40%	48%
			– Parentalidade COMPROMETIDA		50	18	32	100%	100%	100%
			Parentalidade COMPROMETIDA	N	24	10	14	48%	56%	44%
				S	27	8	19	54%	44%	59%
	– S2		– Dor		61	25	36	76%	71%	80%
			Dor	N	34	13	21	56%	52%	58%
				S	28	12	16	46%	48%	44%
			– Parentalidade COMPROMETIDA		76	33	43	95%	94%	96%
			Parentalidade COMPROMETIDA	N	45	19	26	59%	58%	60%
				S	34	15	19	45%	45%	44%
	S2	+ T3	– Dor		29	11	18	78%	69%	86%
			Dor	N	16	6	10	55%	55%	56%
				S	13	5	8	45%	45%	44%
			– Parentalidade COMPROMETIDA		34	14	20	92%	88%	95%
			Parentalidade COMPROMETIDA	N	19	7	12	56%	50%	60%
				S	17	7	10	50%	50%	50%
		+ T4	– Dor		34	14	20	72%	67%	77%
			Dor	N	19	7	12	56%	50%	60%
				S	16	7	9	47%	50%	45%
			– Parentalidade COMPROMETIDA		45	20	25	96%	95%	96%
			Parentalidade COMPROMETIDA	N	27	12	15	60%	60%	60%
				S	19	8	11	42%	40%	44%

[Distrito=Lisboa]

Figura 3.10: Consulta do número de utentes e taxa de prevalência pelas dimensões Tempo, Diagnóstico, Género e Geográfica

e posterior povoamento do DW. Com a terceira fase foi criado o esquema com os cubos multidimensionais necessários, indispensável para o bom funcionamento do servidor OLAP, cujo processo de utilização foi descrito na última fase deste capítulo.

Ao longo de todo o capítulo foi também sendo feita a descrição das ferramentas utilizadas para desenvolvimento de cada fase do projecto.

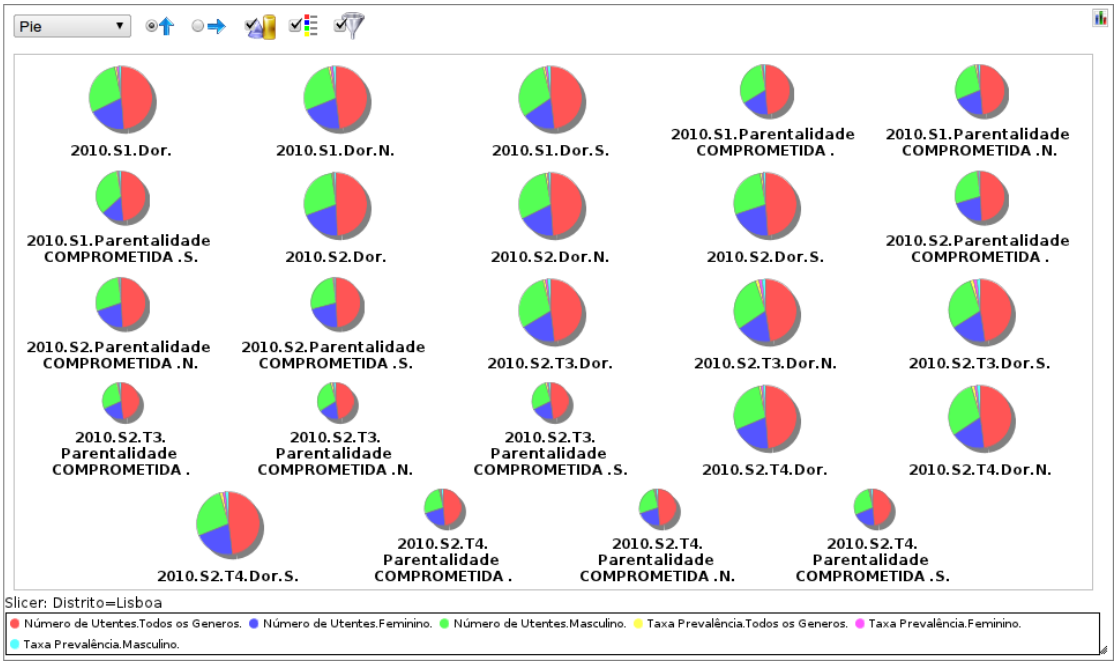


Figura 3.11: Gráfico referente à figura 3.10

Capítulo 4

Avaliação do Modelo

Com a realização deste trabalho, foram implementados vários componentes que permitiram chegar à solução final desejada. Deste modo, foi possível criar situações de análise concretas, testando em termos de funcionalidade e performance o modelo proposto.

Nas primeiras duas secções deste capítulo são efectuadas as avaliações ao sistema, tanto ao nível da eficácia como de performance do mesmo. Posteriormente são feitas algumas observações sobre como melhorar o sistema proposto. Por último, é efectuado um resumo da informação descrita neste capítulo.

4.1 Avaliação da eficácia do sistema

A eficácia da solução OLAP desenvolvida foi testada em diversos momentos pela empresa a quem se destinava, que é também a entidade detentora do repositório fonte dos dados utilizado. Em algumas das sessões de trabalho, representantes da empresa deslocaram-se à Universidade de Évora para efectuar diversos testes ao protótipo desenvolvido, avaliando as suas funcionalidades e fornecendo indicações para a correcta interpretação de dados, necessária para a correcção da solução final.

Diversas alterações à solução foram levadas a cabo, tendo em conta as indicações suge-

ridas. Entre elas, a forma de cálculo das taxas que deram origem às medidas propostas efectuando-se as necessárias alterações para a obtenção dos valores correctos.

4.2 Comparação com Trabalho Relacionado

Não havendo muitos exemplos de trabalhos relacionados na área da saúde, a tentativa de desenvolvimento de modelos multidimensionais já não é propriamente nova, tal como podemos constatar com os trabalhos descritos em 2.3.

A metodologia utilizada em *"A framework for designing a healthcare outcome data warehouse"* [Parmanto et al., 2005] é em muitos pontos semelhante ao modelo proposto neste trabalho. Desde logo começando pela forma de abordagem ao problema, tendo o cuidado de definir os contextos/dimensões necessárias, assim como os diversos níveis de granularidade de cada uma e, ao mesmo tempo, quais as medidas/factos que iriam povoar as tabelas de factos.

Como tal, a metodologia utilizada no presente trabalho coincide com a dos autores anteriormente referidos: o desenvolvimento de um repositório multidimensional baseado num esquema em estrela. Neste caso existem, no entanto, algumas semelhanças e também algumas diferenças nas dimensões criadas, designadamente ao nível das medidas, uma vez que estas são diferentes das adoptadas neste trabalho, na medida em que as fontes de informação são completamente diferentes entre si.

4.3 Avaliação da performance do sistema

O funcionamento do sistema OLAP em questão depende da boa implementação e coordenação de um conjunto de componentes diferentes, tal como é possível constatar através da descrição efectuada no capítulo anterior.

De forma a ser possível avaliar a ferramenta como um todo, foi decidido efectuar diversos testes, através da angariação dos tempos de resposta obtidos pela utilização da interface gráfica.

Como foi referido e demonstrado na secção 3.7, a interface utilizada baseou-se numa aplicação Web. A forma escolhida para efectuar a avaliação da performance, foi a da análise dos tempos de renderização das páginas, mais propriamente das tabelas que representam as consultas efectuadas ao repositório. Desta forma, foi possível avaliar a

performance da ferramenta, de cada vez que é efectuada uma consulta.

4.3.1 Factores de análise

Como qualquer teste realizado, os resultados obtidos estão sempre dependentes de várias componentes e das condições verificadas no momento.

Para realizar a avaliação, foram registados os tempos, em milissegundos(ms), tendo em conta quatro factores diferentes. Os dois primeiros dizem respeito ao número de dimensões seleccionadas e quais as medidas observadas a cada consulta. Os dois restantes foram bastante importantes, pois são os principais responsáveis pela maior ou menor performance da ferramenta. O primeiro é relativo ao número de registos existentes na tabela de factos usada nas consultas e o segundo é relativo à existência ou não de *Cache*, isto é, se a consulta em questão já teria sido executada uma primeira vez ou não, e por sua vez o seu resultado já estaria pré-calculado.

Para o volume de dados foi decidido, numa primeira instância, utilizar-se um número bastante elevado, fixando-se nos 500 mil registos, sendo diminuído para os 100 mil, de seguida para os 10 mil e por último utilizando os dados reais disponíveis, provenientes da base de dados fonte, ou seja, 2960 registos. De referir que à excepção dos dados reais, todos os outros foram gerados automaticamente e distribuídos de forma uniforme pela base de dados, utilizando um *script* adaptado para o efeito a partir de um exemplo pré-existente.

Para obter a melhor fiabilidade possível, para cada conjugação de factores, isto é, cada consulta foi executada três vezes nas mesmas condições, sendo por fim calculado o valor médio e o respectivo desvio padrão através dos tempos diversos angariados. Este processo visou minimizar o impacto de uma eventual influência de factores externos sobre a estatística de tempos.

4.3.2 Resultados obtidos

A tabela 4.1 mostra os tempos que foram obtidos através da execução de três consultas diferentes, tendo em conta os factores descritos na secção 4.3.1. Para este caso foram seleccionadas todas as dimensões disponíveis.

Relativamente às consultas, todas dizem respeito à operação OLAP *drill-down*, dado que

Dimensões	Medidas	#Registos	Cache	1ª Consulta		2ª Consulta		3ª Consulta	
				Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média
Todas	Número Utentes e Taxa de Prevalência	Dados Reais(2960)	Não	4	77	36	194	44	196
			Sim	2	27	5	58	2	63
		10000	Não	13	123	15	413	27	368
			Sim	3	32	22	74	30	99
		100000	Não	13	586	51	3313	23	1610
			Sim	4	31	4	54	12	91
		500000	Não	34	2808	823	12081	57	7340
			Sim	1	25	16	44	8	55

Tabela 4.1: Tempos de resposta com todas as dimensões, em ms

se trata de uma das operações mais exigentes, e por ser também, a par com a operação *drill-up*, das operações mais utilizadas.

A primeira consulta foi efectuada sobre a dimensão *Tempo*, alterando a visualização dos valores totais do ano de 2010 para os seus respectivos semestres. A segunda foi efectuada sobre a dimensão `Diagnostico-Resultado`, sendo possível observar os valores totais para cada diagnóstico específico, relativamente ao ano de 2010. A terceira consulta foi efectuada sobre a dimensão *Geográfica* sendo possível observar os valores totais respeitantes a cada distrito, relativamente ao diagnóstico `Parentalidade Comprometida` e ao ano de 2010. O código MDX correspondente às consultas efectuadas está disponível para consulta no anexo A.

Com base na observação da tabela 4.1 podemos constatar que, como era de esperar, à medida que o número de registos foi aumentando, o respectivo tempo das consultas foi aumentando também, atingindo o valor máximo de 12081 ms aquando da realização da segunda consulta.

De facto, em todos os testes efectuados sem a utilização de *Cache*, a segunda consulta foi sem dúvida a mais demorada, exceptuando o caso dos dados reais, embora neste a diferença não seja significativa relativamente à consulta seguinte. Esta demora deve-se principalmente à existência do membro calculado, cujo processamento dá origem aos valores da *Taxa de Prevalência*. Como referido em 3.6.2, a sua fórmula está directamente relacionada com a dimensão que é usada na operação *drill-down* efectuada na segunda consulta. Dado que se trata de uma taxa, os seus valores variam de forma dinâmica à medida que o número de utentes relativos a um diagnóstico varia. Desta forma, o seu cálculo não poderia de maneira alguma ser efectuado no processo de ETL, como foi pensado numa primeira abordagem ao problema. Posto isto, nas consultas sobre esta

dimensão, existe uma quantidade bastante elevada de cálculos a realizar para que seja apurado o valor taxa, referente a cada contexto, e para que este seja ao mesmo tempo armazenado em *Cache*. Este processo dá origem a uma perda de performance de cerca de 2000 ms face à mesma consulta, sem a presença deste membro calculado.

A tabela da dimensão `Tempo` é por norma a que possui maior volume de dados, o que poderia levar a pensar que o tempo de execução de uma consulta sobre esta seria maior. Tal não se verifica e, com a observação dos resultados contidos na tabela 4.1, podemos constatar que foi exactamente o contrário, sendo a consulta mais demorada de 2782 ms, quando efectuada sobre os 500 mil registos. Tal deve-se ao facto de esta dimensão ter um comportamento especial pois, uma vez que o nível mais alto na hierarquia foi definido como `Ano`, os resultados obtidos são, por defeito, agrupados pelo ano em que ocorreram e não pelo total. Deste modo, existe um cálculo prévio dos resultados relativamente à dimensão `Tempo`, resultando numa menor duração para a execução das consultas.

Relativamente à terceira consulta, era expectável que o tempo de execução da mesma fosse menor que a anterior, dado que se tratava de uma consulta menos abrangente, aos resultados por distrito correspondentes a um diagnóstico especifico no ano de 2010. Tal acabou mesmo por se verificar, exceptuando o caso dos dados reais, mas como estamos a falar de um tempo de execução inferior a 500 ms, este não é sequer perceptível pelo utilizador.

Num segundo teste, uma vez que nas operações de *drill-down* apenas eram utilizadas três das cinco dimensões existentes, quisemos saber se o facto de estarem seleccionadas duas dimensões, sem serem efectuadas quaisquer operações sobre elas, faria aumentar ou diminuir o nível de performance da ferramenta. Como tal, neste segundo teste foram executadas as mesmas três consultas, mas apenas foram seleccionadas as três dimensões utilizadas nas mesmas.

Com os resultados obtidos pela tabela 4.2 pudemos constatar que, apesar de haver um melhoramento em todos os tempos de execução, este não foi de modo algum significativo, sendo, no máximo, de pouco mais de 500 ms, no caso da terceira consulta e na presença do maior número de registos na tabela de factos. De qualquer forma, foi perceptível que quanto menos dimensões forem seleccionadas melhor é o desempenho da ferramenta.

Por último, tanto na tabela 4.1 como na 4.2 foi possível constatar que a presença de *Cache* permite um ganho significativo de performance na utilização da ferramenta. Não só para a repetição de consultas anteriormente efectuadas, mas também para novas. Mui-

Dimensões	Medidas	#Registos	Cache	1ª Consulta		2ª Consulta		3ª Consulta	
				Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média
Tempo Diagnóstico Geográfica	Número Utentes e Taxa de Prevalência	Dados Reais(2960)	Não	1	60	22	163	23	149
			Sim	4	22	5	31	4	30
		10000	Não	6	110	13	386	2	273
			Sim	2	25	5	39	10	46
		100000	Não	10	575	89	3245	73	1518
			Sim	5	21	8	34	3	29
		500000	Não	13	2782	246	11469	54	6450
			Sim	4	23	6	37	1	42

Tabela 4.2: Tempos de resposta apenas com as três dimensões necessárias, em ms

tas vezes, ao criar uma nova consulta, verifica-se a utilização de dados anteriormente agregados e já presentes em *Cache*. Este facto permite também aumentar o nível de performance da ferramenta, diminuindo o tempo necessário para a execução das consultas.

4.4 Melhoramentos

A utilização da *Cache* agiliza o processo de execução de consultas, tornando-o constantemente mais rápido à medida que se navega pelos dados e que vão sendo efectuadas mais agregações.

Com base na análise das tabelas 4.1 e 4.2, mais precisamente nos tempos relativos à segunda e terceira consulta, sem a presença de *Cache* é perceptível que quanto mais detalhada é a informação, menos tempo demora a executar a consulta. Isto é, ao efectuarmos a terceira consulta estamos a analisar os valores das medidas relativamente a um diagnóstico específico e não sobre todos os diagnósticos, sendo o tempo de execução bastante mais baixo, dado que é necessário um menor número de agregações.

Uma das formas de melhorar a performance da ferramenta é a utilização das vantagens que a *Cache* traz à ferramenta, ou seja, determinar as consultas mais demoradas, que por norma são as que se referem a operações efectuadas sobre os níveis mais altos das hierarquias das dimensões, e executá-las na inicialização do servidor. Desta forma, parte das agregações e dos cálculos já estarão armazenados em *Cache*, baixando os tempos das consultas para valores semelhantes aos representados nas tabelas 4.1 e 4.2, quando estamos perante a existência de *Cache*, ou seja, valores inferiores a 1000 ms. Ao mesmo tempo, este melhoramento torna a execução das consultas posteriores mais rápida, pois algumas agregações já estarão armazenadas, sendo o processo optimizado.

Uma das consultas candidatas a esta forma de melhoramento é precisamente a segunda consulta utilizada nos testes e descrita em 4.3.2, assim como a mesma consulta mas desta vez efectuada sobre a dimensão `Geográfica`, uma vez que é uma das dimensões mais povoadas.

4.5 Resumo

Este capítulo foi dedicado à descrição e realização de diversos testes, sobre o modelo proposto no capítulo anterior.

Desta forma, na secção 4.1 foi descrita a forma como o modelo foi sendo avaliado ao nível da eficácia e precisão dos resultados obtidos, realizando também a comparação da metodologia utilizada com trabalhos relacionados

Na segunda secção foi feita uma avaliação ao modelo em termos de performance, sendo em 4.3.1 efectuada a explicação dos factores utilizados na realização dos testes e posteriormente em 4.3.2 apresentados e analisados os resultados obtidos. Por último, em 4.4 são sugeridos alguns melhoramentos possíveis, de modo a conseguir optimizar os resultados obtidos na secção anterior.

No capítulo seguinte serão feitas as considerações finais tendo em conta os resultados descritos.

Capítulo 5

Conclusões

Depois de apresentado o trabalho realizado, através da descrição da metodologia utilizada em cada fase do trabalho e ao mesmo tempo apresentando alguns exemplos da solução final, há que efectuar um balanço do mesmo. No presente capítulo é feita, pois, uma análise do trabalho, sendo na secção 5.1 efectuado não só um recapitular dos objectivos alcançados, mas também dos diversos componentes do trabalho. Na secção 5.2 são enumerados alguns aspectos em termos de trabalho futuro.

5.1 Objectivos Alcançados e contribuições

A solução apresentada neste trabalho engloba-se na temática de Business Intelligence, tendo sido o seu principal foco a implementação de uma solução de OLAP para a área da saúde.

Tratando-se de uma temática bastante ampla e que engloba diferentes componentes, foi feita uma descrição mais teórica de cada um destes, de modo a facilitar o seu entendimento, tendo em vista o trabalho desenvolvido.

Levando em conta a necessidade de análise dos registos de enfermagem, contidos num sistema de gestão de base de dados, por meio de três taxas distintas em diversos con-

textos, o processo de desenvolvimento passou em grande parte pela implementação e povoamento de um DW, adequado às necessidades do problema.

O trabalho foi desenvolvido em diversas fases, sendo que o método seguido em cada uma poderá ter alguma relevância e servir como base a futuros sistemas nesta área. As fases de desenvolvimento foram as seguintes:

1. Construção de um repositório multidimensional com base numa constelação de factos;

2. Povoamento do repositório por meio de diversas transformações, constituindo o processo de ETL;

3. Construção de esquema com informação sobre os cubos, responsável por permitir ao servidor OLAP efectuar as traduções entre o código multidimensional MDX e o código SQL reconhecido pelo repositório;

4. Utilização de ferramenta de visualização STPivot sobre um servidor ROLAP;

5. Construção de consultas pré-definidas sobre a ferramenta de visualização.

Os objectivos traçados no início do trabalho foram atingidos, tendo os valores correspondentes às medidas propostas e os contextos em que se inserem, devolvidos pela ferramenta de OLAP, sido validados por dois responsáveis da empresa contratante.

A importância da construção de um bom repositório multidimensional está bem patente neste trabalho. Dado que este tipo de soluções é bastante utilizado na área dos negócios, a sua implementação para outras áreas torna-se mais difícil devido a alguma falta de exemplos práticos.

Desta forma, este trabalho constitui um contributo relevante, por um lado, na forma de construção de um repositório multidimensional de análise de dados na área da saúde. Por outro lado, apresenta uma solução funcional de um sistema de OLAP que fornece um conjunto de operações que permitem a leitura de forma intuitiva e organizada dos dados.

5.2 Trabalho Futuro

A abordagem metodológica e técnica deste trabalho foi condicionada por diversos factores, nomeadamente o objecto do estudo (registos no domínio da enfermagem pediátrica) e a consideração de que a ferramenta OLAP utilizada era a que melhor se adaptava para alcançar os objectivos pretendidos.

Naturalmente que a evolução deste mesmo trabalho ou novas necessidades que nele se venham a verificar, determinarão a utilização de outras ferramentas e outras metodologias. Quer seja na procura de melhores formas de disposição dos dados ao nível do repositório, quer seja ao nível do processo de ETL, ou mesmo outras condições de análise. Neste contexto, poderão existir alterações ao nível de:

- Repositório multidimensional: o esquema da base de dados relacional associado a este repositório poderá beneficiar de optimizações, de modo a melhorar o desempenho da solução. Após alguns meses de funcionamento, poder-se-á fazer um ajuste nos índices em função do padrão de crescimento dos dados;
- Outros sistemas de gestão de base de dados: testar a solução recorrendo a outros sistemas de gestão, como por exemplo o PostgreSQL ou o Oracle. Desta forma será possível constatar qual o sistema com melhor desempenho para a solução apresentada;
- Processo de ETL: a criação de um processo automático de utilização das transformações desenvolvidas para povoar o repositório multidimensional, de modo a optimizar o processo, tornando possível efectuar agendamentos periódicos;
- Interface: ao nível da interface, a margem para alterações é bastante grande devido à constante inovação das aplicações web. Mesmo já possuindo um grande número de operações, a inclusão de novas trará mais e melhores condições de análise;
- Avaliação: a solução final necessita de alguns testes por parte de utilizadores comuns, de modo a ser possível encontrar falhas e desta forma tornar o sistema mais robusto.
- Hadoop e Hive: numa perspectiva a longo prazo, outra das hipóteses de incremento significativo da performance será a utilização da plataforma Hadoop[1] de modo a conseguir distribuir o processamento de grandes volumes de dados por diversas

[1] http://hadoop.apache.org

máquinas, ou seja, criando um *cluster*. Esta plataforma, aliada ao software Hive[2], que fornece uma linguagem de consultas designada por *Hive Query Language*, vai permitir executar consultas aos dados distribuídos por este mesmo cluster.

Deste modo, variando as condições, a margem de evolução do trabalho desenvolvido é bastante grande de acordo com outras medidas passíveis de análise, novos contextos/dimensões e âmbitos mais abrangentes.

Contudo, o presente trabalho assume uma importância apreciável enquanto paradigma para outras especialidades médicas, que não exclusivamente a pediatria, mediante as necessárias adaptações.

[2]http://hive.apache.org

Bibliografia

[Bhole, 2010] Bhole, G. (2010). Building a data mart using star schema. Master's thesis, San Diego State University.

[Bouman and Dongen, 2009] Bouman, R. and Dongen, J. (2009). *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*. Wiley Publishing, Inc.

[Chaudhuri and Dayal, 1997] Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and olap technology. *Association for Computing Machinery, Inc.*

[Gunderloy and Sneath, 2001] Gunderloy, M. and Sneath, T. (2001). *SQL Server Developer's Guide To OLAP With Analysis Services*. Sybex.

[Han and Kamber, 2000] Han, J. and Kamber, M. (2000). Data mining: Concepts and techniques. Master's thesis, Simon Fraser University.

[Inmon, 1992] Inmon, W. H. (1992). *Building the Data Warehouse*. QED Technical Publishing Group, Wellesley, Massachusetts.

[Jedox,] Jedox. Palo plan analyse report. `http://www.jedox.com/pt/produtos/produtos-oferecidos.html`.

[Kimball and Caserta, 2004] Kimball, R. and Caserta, J. (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley Publishing, Inc.

[Kimball and Ross, 2002] Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: 2nd Edition The Complete Guide to Dimensional Modeling*. Wiley Computer Publishing.

[Luján-Mora, 2005] Luján-Mora, S. (2005). Data warehouse design with uml. Master's thesis, Universidad de Alicante.

[Maddi and Khan, 2007] Maddi, S. R. and Khan, V. (2007). Comparative analysis of on-line analytical processing tools. Master's thesis, IT University of Goteborg.

[Malinowski and Zimányi, 2006] Malinowski, E. and Zimányi, E. (2006). Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data & Knowledge Engineering*.

[Moulton,] Moulton, F. C. Olap and olap server definitions. http://www.moulton.com/olap/olap.glossary.html. Acedido em Setembro de 2011.

[MySQL,] MySQL. Mysql - the world's most popular open source database. http://www.mysql.com/.

[Parmanto et al., 2005] Parmanto, B., Scotch, M., and Ahmad, S. (2005). A framework for designing a healthcare outcome data warehouse. *Em Perspectives in Health Information Management*.

[Pearson, 2002] Pearson, W. (2002). Mdx at first glance: Introduction to sql server mdx essentials. *Database Journal: The Knowledge Center for Database Professionals*.

[Pentaho, a] Pentaho. Mondrian schema workbench. http://mondrian.pentaho.com/documentation/workbench.php.

[Pentaho, b] Pentaho. Pentaho bi platform / server. http://community.pentaho.com/projects/bi_platform/.

[Pentaho, c] Pentaho. Pentaho kettle project. http://kettle.pentaho.com/.

[Pentaho, d] Pentaho. Pentaho mondrian project. http://mondrian.pentaho.com/.

[Pentaho, e] Pentaho. Pentaho open source business intelligence. http://www.pentaho.com.

[Puhr, 2002] Puhr, C. (2002). The clinical data warehouse. Master's thesis, Medical University of Vienna.

[Reyes, 2010] Reyes, E. P. (2010). A systems thinking approach to business intelligence solutions based on cloud computing. Master's thesis, Massachusetts Institute of Technology.

[SQLPOWER,] SQLPOWER. Sql power arquitect. http://www.sqlpower.ca/page/architect.

[StrateBI,] StrateBI. Stpivot jpivot with steroids. http://code.google.com/p/stpivot/.

[Taleb, 2011] Taleb, A. (2011). Query optimization and execution for multi-dimensional olap. Master's thesis, Concordia University.

[Tam, 1998] Tam, Y. J. (1998). Datacube: Its implementation and application in olap mining. Master's thesis, Simon Fraser University.

[Vassiliadis and Simitsis, 2009] Vassiliadis, P. and Simitsis, A. (2009). Extraction, transformation, and loading. *Em Encyclopedia of Database Systems*.

[Westerlund, 2008] Westerlund, P. (2008). Business intelligence: Multidimensional data analysis. Master's thesis.

Anexos

Anexo A

Código MDX

Este anexo contém o código MDX das três consultas executadas para obter os tempos necessários, de modo a realizar o processo de avaliação do modelo proposto. Este complementa a informação fornecida na secção 4.3.2

1ª consulta

```
select
  NON EMPTY {[Measures].[Numero Utentes], [Measures].[Taxa Prevalencia]}
    ON COLUMNS,
  NON EMPTY Crossjoin(Hierarchize(Union({[Tempo.Tempo Anual].[2010]},
  [Tempo.Tempo Anual].[2010].Children)),{([Diagnostico-Resultado].[Todos],
  [Geografica].[Total Agregado],[Idades.Ate 24 Meses].[Meses],
  [Genero].[Todos os Generos])})
    ON ROWS
from [Taxa Prevalencia]
```

2ª consulta

```
select
  NON EMPTY {[Measures].[Numero Utentes], [Measures].[Taxa Prevalencia]}
    ON COLUMNS,
  NON EMPTY Crossjoin(Hierarchize(Union(Union(Crossjoin(
    {[Tempo.Tempo Anual].[2010]},
    {[Diagnostico-Resultado].[Todos]}),Crossjoin({[Tempo.Tempo Anual].[2010]},
     [Diagnostico-Resultado].[Todos].Children)),
      Crossjoin([Tempo.Tempo Anual].[2010].Children,
    {[Diagnostico-Resultado].[Todos]}))), {([Geografica].[Total Agregado],
     [Idades.Ate 24 Meses].[Meses], [Genero].[Todos os Generos])})
    ON ROWS
from [Taxa Prevalencia]
```

3ª consulta

```
select
  NON EMPTY {[Measures].[Numero Utentes], [Measures].[Taxa Prevalencia]}
    ON COLUMNS,
  NON EMPTY Crossjoin(Hierarchize(Union(Union(Union(Crossjoin(
    {[Tempo.Tempo Anual].[2010]},
    {([Diagnostico-Resultado].[Todos], [Geografica].[Total Agregado])}),
    Crossjoin({[Tempo.Tempo Anual].[2010]},
    Crossjoin([Diagnostico-Resultado].[Todos].Children,
    {[Geografica].[Total Agregado]}))), Crossjoin({[Tempo.Tempo Anual].[2010]},
    Crossjoin({[Diagnostico-Resultado].[Parentalidade COMPROMETIDA ]},
    [Geografica].[Total Agregado].Children))),
    Crossjoin([Tempo.Tempo Anual].[2010].Children,
    {([Diagnostico-Resultado].[Todos], [Geografica].[Total Agregado])}))),
    {([Idades.Ate 24 Meses].[Meses], [Genero].[Todos os Generos])})
    ON ROWS
from [Taxa Prevalencia]
```