

# Uma Ferramenta OLAP e um Data Mart Comercial: Uma Aplicação Prática na Indústria Calçadista utilizando *softwares open source*

**Edmilson José Wonter Felber**

Safetech Informática LTDA, Rua Carlos Biehl, 92, Sala 9 - Centro,  
Sapiranga, RS 93800-000, Brasil  
edmilson@safetech.com.br

e

**Juliano Varella de Carvalho**

GPTI, Centro Universitário FEEVALE  
Novo Hamburgo, RS 93352-000, Brasil  
julianovc@feevale.br

## Abstract

In the current economic context, where companies competition is each time more related to its capacity to transform information into knowledge and after to transform this knowledge in decisions and action of business, Data Warehouse (DW) has a basic paper as tool of support to the processes of decision-making. However, to take off advantage of information's resources of satisfactory form, it's necessary that the companies optimize and give more emphasis in the quality to the process of decision-making. In this context, tools and concepts are used to organize the business-oriented information of the companies. Beyond DW, other important concepts are Data Mart (DM), Business Intelligence (BI), dimensional modeling and OLAP tools, that constitute the strategical pillars of the Decision Support System (DSS). This paper shows a DM creation and a OLAP tool for information's analysis of a shoe factory commercial department. This solution was based on free and/or Open Source software, as the Mondrian and the Jpivot, and shows that this tools are very efficient in the process of decision-making.

**Keywords:** Databases, Web Systems, Data Mart, OLAP, Data Warehouse, Mondrian, JPivot

## Resumo

No contexto econômico atual, onde a competição entre as empresas está cada vez mais relacionada à sua capacidade de transformar informação em conhecimento e este último em decisões e ações de negócio, o *Data Warehouse* (DW) exerce papel fundamental como ferramenta de apoio aos processos de tomada de decisão. Contudo, para que se possa tirar vantagem dos recursos de informação de forma satisfatória, é cada vez mais imperativo que as empresas otimizem e dêem mais ênfase à qualidade durante o processo de tomada de decisão. Neste contexto, são empregadas ferramentas e conceitos que organizam as informações de negócios das empresas. Dentre estes, além do DW, destacam-se também, *Data Mart* (DM), *Business Intelligence* (BI), modelagem dimensional e ferramentas OLAP, que constituem os pilares estratégicos dos Sistemas de Apoio a Decisão (SAD). Este artigo apresenta a criação de um DM e uma ferramenta OLAP para a análise das informações do departamento comercial de uma indústria calçadista. Toda a solução é baseada em softwares de distribuição gratuita e/ou *Open Source*, como o *Mondrian* e *Jpivot* e, desta forma, demonstra como estes podem tornar-se ferramentas de grande eficiência ao processo de tomada de decisão.

**Palabras claves:** Bancos de dados, Sistemas Web, Data Mart, OLAP, Data Warehouse, Mondrian, JPivot

# 1 INTRODUÇÃO

Desde o início da era computacional, as organizações têm usado os dados de suas bases operacionais para atender as necessidades de informações. Uma das grandes dificuldades das empresas é o gerenciamento dos dados oriundos de diversos sistemas do ambiente operacional. Grandes bases de dados são criadas pelo acúmulo de informações resultantes de operações transacionais, pouco aproveitadas pelos responsáveis nas tomadas de decisões, devido a complexidade de extração das mesmas. Agrupar essas informações, interpretá-las e tirar conclusões, não é uma tarefa fácil. É preciso extrair de cada base de dados, as informações que realmente interessam e padronizá-las para que possam ser analisadas.

Conforme [4], os sistemas de produção de uma empresa recuperam e atualizam um registro por vez, normalmente atendendo a muitos usuários de forma concorrente, exigindo também um tempo de resposta muito baixo. Já os Sistemas de Apoio a Decisão (SAD), usualmente lidam com poucos usuários por vez e os requisitos em termos de tempo de resposta não são críticos. No entanto, geralmente lidam com consultas complexas, não antecipadas ou previstas, envolvendo grande quantidade de registros. Os SADs visam atender as necessidades dos executivos de uma organização e, geralmente, os resultados das pesquisas são demonstrados através de gráficos e tabelas, apresentando informações com alto índice de valor agregado.

O termo SAD é utilizado cada vez com menos frequência, em seu lugar, tem sido cada vez mais usual o termo *Business Intelligence* (BI). Este termo é um conceito empregado a ferramentas, tecnologias e metodologias, que tem como objetivo fornecer informações estratégicas, que apóiam a tomada de decisão. Segundo [2], BI “representa a habilidade de se estruturar, acessar e explorar informações, normalmente guardadas em DW/DM (*Data Warehouse/Data Mart*), com o objetivo de desenvolver percepções, entendimentos, conhecimentos, os quais produzirão uma melhor tomada de decisão”. A Figura 1 mostra os principais componentes deste ambiente.

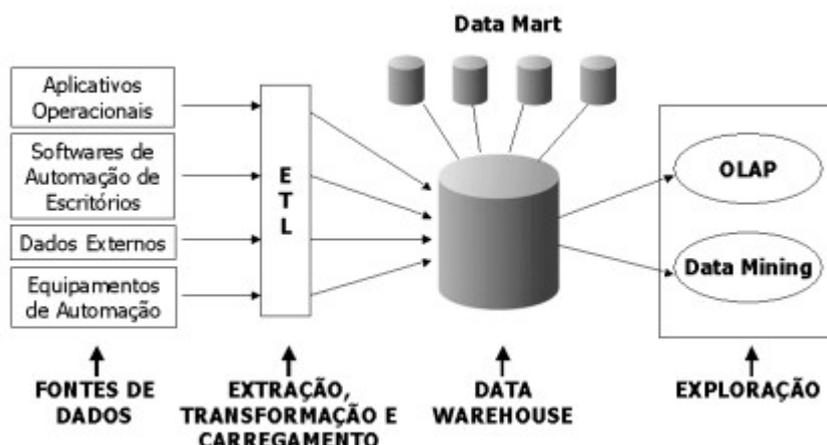


Figura 1 Componentes de um ambiente

Contudo, uma das características das soluções proprietárias das ferramentas de BI é o alto custo da implementação destas, o que, na maioria das vezes, torna proibitivo a sua utilização em pequenas, médias e em certas situações até em grandes empresas. De acordo com [5], este universo está começando a ser descoberto pelo movimento *Open Source*, fato esse que fornecerá ferramentas para que os desenvolvedores de aplicações ofereçam a seus clientes soluções de gestão de informação com qualidade, a um custo razoável.

O trabalho explanado neste artigo desenvolveu um Sistema de Apoio a Decisão composto por um DM e uma ferramenta OLAP para a análise das informações do departamento comercial de uma indústria calçadista, utilizando ferramentas de distribuição gratuita e/ou *Open Source*.

### 1.1 Data Warehouse

A idéia do DW é armazenar os dados em vários graus de relacionamento e sumariação, de forma a agilizar os processos de tomada de decisão por diferentes níveis gerenciais. Esses dados, oriundos de sistemas de informação de produção, deverão estar integrados e disponíveis, permitindo diversas formas de consultas, através dos mecanismos amigáveis das ferramentas de usuários [2].

Segundo [8], “Data warehouse é um conjunto de dados baseado em assuntos, integrado, não-volátil, e variável em relação ao tempo, de apoio às decisões gerenciais”.

De acordo com [14], entende-se por “baseado em assuntos”, que o DW foca as principais entidades do negócio; integrado, denota que os dados estão armazenados em formato consistente, ou seja, especificando convenções, restrições de domínio, atributos físicos e medições coerentes; a característica da não volatilidade diz respeito aos dados não se alterarem depois de incluídos no DW, pois os novos dados são adicionados a ele, integrando-se com àqueles previamente armazenados. Por fim, ser variante no tempo significa que os dados estão associados a um ponto no tempo, como por exemplo, ano, semestre, mês, trimestre, período de pagamento.

“A granularidade diz respeito ao nível de detalhe ou de resumo contido nas unidades de dados existentes no *data warehouse*. Quanto mais detalhe, mais baixo o nível de granularidade. Quanto menos detalhe, mais alto o nível de granularidade.” [7].

À medida que o nível de granularidade aumenta, o número de consultas que podem ser atendidas diminui, sendo que em uma granularidade mínima as consultas mais detalhadas podem ser respondidas. Portanto, é necessário encontrar um ponto de equilíbrio. O nível adequado de granularidade deve ser definido de tal forma que atenda as necessidades do usuário, tendo como limitação os recursos disponíveis [6]. Diante disto, a definição da granularidade de dados é a etapa mais importante do projeto de um DW, porque ela afeta profundamente o volume de dados dentro dele e, ao mesmo tempo, afeta o tipo de consulta que pode ser atendida. Devem-se definir níveis adequados de granularidade, de acordo com as necessidades do usuário.

### 1.2 Data Mart

A criação de um DW requer tempo, capital e um considerável esforço gerencial. Por estes motivos, em diversas situações, as empresas iniciam o projeto de DW focando nas necessidades de áreas específicas dentro da organização. Estas estruturas menores de armazenamento de dados são chamadas *Data Marts* (DM). Um DM é um pequeno DW que fornece suporte a um número reduzido de pessoas, com uma visão mais especializada e limitada dos dados.

Atualmente, diversas empresas optam pelo DM por causa do seu custo mais baixo e do tempo menor de implementação da solução e, além disto, estes podem servir como um laboratório de teste àquelas empresas que desejam investigar e explorar os benefícios do DW.

### 1.3 ETL

No ambiente de DW/DM, os dados são inicialmente extraídos de sistemas operacionais e de fontes externas, posteriormente são integrados e transformados antes de serem carregados no DW. A ETL (extração, transformação e carga dos dados), também denominada *Data Staging Area*, é uma etapa

crítica da construção de um DW/DM, pois envolve a movimentação dos dados que posteriormente serão analisados. Segundo [9], esta etapa, diversas vezes, consome o maior período do processo de implantação de um DW/DM.

Uma vez que os dados tenham sido extraídos, um conjunto de transformações deve ser processado sobre esses dados. A transformação dos dados pode ser simples ou complexa, dependendo da natureza das fontes. Em algumas situações, múltiplos estágios de transformações são necessários e, dentre estes estágios, dados são limpos, eliminados, combinados, validados, consolidados, agregados e sumariados.

A última etapa deste processo, a carga de dados, envolve a exportação destes, transformados, para o DW/DM. Tal fase deve manter a integridade dos dados, executar um carga incremental ou total, e por fim otimizar a operação.

O artigo é composto de sete seções. A primeira delas dá uma visão geral sobre as tecnologias envolvidas no processo de tomada de decisão. A segunda seção foca nas diferenças entre modelagem relacional e modelagem dimensional. A próxima seção comenta sobre as principais características da tecnologia OLAP. A quarta e quinta seções são dedicadas às ferramentas *open-source* utilizadas no estudo de caso proposto. A sexta seção deste artigo ilustra como foi desenvolvido o sistema resultante, e a conclusão deste encontra-se na última seção.

## 2 MODELAGEM DIMENSIONAL

Nos bancos de dados relacionais, a redundância dos dados é evitada, sendo eliminada através de processos de normalização. Em ambientes como DW/DM, as transações operam sobre um grande volume de dados e não são simples nem freqüentes, tornando o processo normalização das tabelas inconveniente [13]. Um outro ponto que distingue o banco de dados relacional do DW/DM está relacionado à modelagem dos dados. Enquanto o banco de dados relacional geralmente utiliza a modelagem Entidade-Relacionamento (ER)<sup>1</sup>, o DW/DM utiliza-se de uma modelagem lógica chamada de modelagem dimensional, técnica que permite uma visão multidimensional dos dados.

“Como o modelo relacional trabalha com normalização, suas tabelas possuem menos registros e não têm redundâncias, apresentando assim uma melhor performance nas tarefas do dia a dia, como inclusões, alterações e exclusões de registros, mas ele só é adequado para consultas simples de poucos registros. Para análises mais complexas com um universo de registros maior, o modelo dimensional oferece uma melhor alternativa, economizando em junções com várias tabelas, e armazenando dados que facilitam a análise das informações” [6]. O modelo dimensional é composto pela tabela de fatos e pelas tabelas de dimensões [1].

### 2.1 Fatos e Dimensões

Segundo [10], a tabela de fatos é a principal tabela de um modelo dimensional, onde as medições numéricas de interesse da empresa estão armazenadas. A palavra *fato* é usada para representar uma medição de negócio, como quantidades, valores e indicadores. A tabela de fatos registra os fatos que serão analisados. É composta por uma chave primária (formada por uma combinação única de valores de chaves de dimensão) e pelas métricas de interesse para o negócio.

<sup>1</sup> Proposta por Peter P. Chen em 1976, A modelagem Entidade-Relacionamento (ER) identifica os elementos de importância na organização (entidades), as propriedades destes elementos (atributos) e como eles estão relacionados uns aos outros (relacionamentos). O modelo resultante da informação é independente de qualquer armazenamento de dados ou método de acesso.

De acordo com [6], a tabela de fatos é sempre esparsa, ou seja, possui um número relativamente pequeno de todas as combinações possíveis de valores de chaves. Por exemplo, no caso de um banco de dados de uma área comercial, a presença de todas as combinações possíveis representaria que todos os clientes compram para todas as datas de entrega, todos os produtos da empresa, o que é praticamente impossível. Por isso podemos concluir que esse banco é extremamente esparsa, pois uma porcentagem muito pequena de todas as combinações possíveis de clientes, produtos e datas de entrega aparecerá nele.

Uma tabela de dimensão contém as descrições textuais do negócio. Seus atributos são fonte das restrições das consultas, agrupamento dos resultados, e cabeçalhos para relatórios. As dimensões são os aspectos pelos quais se pretende observar as métricas relativas ao processo que está sendo modelado. Cada dimensão deve ser identificada com uma única chave primária. Essa chave é a base da integridade referencial no relacionamento com a tabela de fatos. Uma tabela de dimensão é composta também de atributos; os melhores atributos são textuais e discretos, e devem consistir de palavras reais, evitando-se o uso de códigos e abreviações [10]. Esses atributos descrevem as linhas na tabela de dimensão.

## 2.2 Agregados

Devido ao grande volume de dados envolvidos nas consultas para análise de dados, o tempo de resposta pode se tornar um fator crítico. Uma técnica utilizada para obter ganhos de desempenho é a criação de tabelas agregadas. Basicamente, consiste em criar novas tabelas com os dados da tabela de fatos, mas alterando a granularidade da mesma, gerando assim tabelas menores, com dados sumariados [6].

Há de se ter cautela na criação de agregados pois, de certa forma, eles contrariam os conceitos estabelecidos de não-redundância para os banco de dados. Além disso, ocupam mais espaço para o armazenamento dos dados em um estado pré-processado [2].

## 2.3 Cubos

Cubo de dados é uma estrutura multidimensional que expressa a forma na qual os tipos de informações se relacionam entre si. É formado pela tabela de fatos e pelas tabelas de dimensão que a circundam e representam possíveis formas de visualizar e consultar os dados [6]. Por exemplo, um cubo contendo informações de vendas poderá ser composto pelas dimensões tempo, região, produto, cliente, fornecedores e medidas. Medidas típicas seriam, por exemplo, valor de venda, unidades vendidas, custos, margem de lucro, etc.

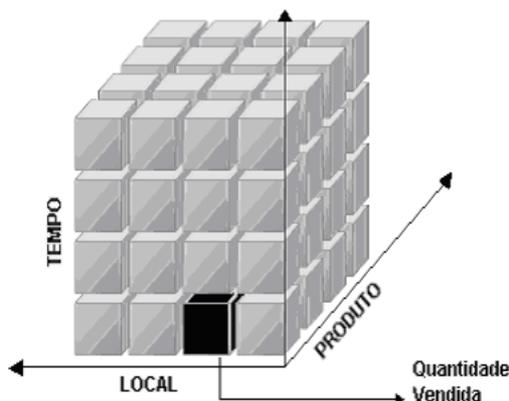


Figura 2: Cubo de dados.

A Figura 2 mostra um exemplo de um cubo de dados, contendo as dimensões tempo, produto e local, sendo quantidade vendida uma de suas medidas.

### 3 OLAP

Dentre as ferramentas envolvidas na apresentação dos dados de um DW/DM, destaca-se a tecnologia denominada OLAP - *On-line Analytical Processing*, que se caracteriza por transformar dados em informações eficientes, dinâmicas e flexíveis. O Conselho OLAP<sup>2</sup>, define esta tecnologia como “a categoria da tecnologia de software que permite que analistas, gerentes e executivos obtenham, de maneira rápida, consistente e interativa, acesso a uma variedade de visualizações possíveis de informações que foi transformada de dados puros para refletir a dimensão real do empreendimento do ponto de vista do usuário” [7].

O objetivo de uma ferramenta OLAP é transformar dados em informações capazes de dar suporte a decisões gerenciais de forma flexível e em tempo hábil. De acordo com [14], as ferramentas OLAP permitem ao usuário, navegar entre diferentes granularidades de um cubo de dados. Para isto, são utilizadas operações que procuram responder inúmeras questões gerenciais, dentre estas operações destacam-se: *Drill-across*, *Drill-down*, *Drill-up*, *Slice-dice*, *Drill-through*, *Pivoting* e *ad-hoc* [15]. São elas as responsáveis por permitir a navegação do usuário nas tabelas de fatos e dimensões existentes.

### 4 MONDRIAN

*Mondrian* é um servidor OLAP *Open Source* escrito em linguagem *Java*, responsável por executar consultas escritas em linguagem MDX (*Multidimensional Expressions*). Ele lê os dados de um RDBMS (*Relational Databases Management System*), e apresenta o resultado em um formato multidimensional através de uma API *Java*. A característica de realizar consultas OLAP em dados armazenados em um banco de dados relacional e retornar consultas em forma multidimensional classifica o *Mondrian* como um software ROLAP (Relational OLAP) [3].

A linguagem de consulta MDX é semelhante à linguagem SQL, porém exclusiva para consultas multidimensionais em bancos de dados. Ela foi originalmente criada pela *Microsoft* para utilização com o produto *SQL Server OLAP Services*, como parte da especificação OLE DB/OLAP API [11]. Ela é a linguagem de consulta implementada pelo *Mondrian*.

Na estrutura do *Mondrian*, basicamente três tipos de dados precisam ser armazenados: dados das tabelas fato, dimensões e agregados. Agregados pré-computados são necessários para grandes conjuntos de dados, caso contrário, certas consultas não poderiam ser executadas sem que fosse efetuada uma leitura completa do conteúdo da tabela de fatos.

#### 4.1 Mondrian Schema

De acordo com [3], a lógica do *Mondrian* é implementada através de *Schemas*, que definem o modelo multidimensional lógico e o mapeamento deste modelo em um modelo físico e relacional. O modelo lógico consiste de elementos definidos pelo *Schema*. Estes elementos são: cubos, dimensões, hierarquias, níveis e membros. O modelo físico é a fonte de dados que é mapeada pelo modelo lógico através do *Schema*. Tipicamente, é um banco de dados onde as informações ficam armazenadas em tabelas relacionais.

---

<sup>2</sup> O Conselho OLAP (*OLAP Council*) é uma organização sem fins lucrativos, patrocinada por vários fornecedores de ferramentas OLAP, cujo objetivo é promover a difusão da tecnologia OLAP.

Conforme [Brito 2004], a representação destes *Schemas* é feita através de arquivos XML (*eXtensible Markup Language*), utilizando os mesmos conceitos relacionados à análise dimensional: cubo (*cube*), representando, em alto nível, a lógica multidimensional do sistema, bem como os fatos (*measures*) e dimensões (*dimensions*). Atualmente, uma das maneiras de criar um *Schema* é editar manualmente o arquivo XML em um editor de texto. A Figura 3 exibe um exemplo de arquivo XML definindo um *Schema*.

```
<Schema>
  <Cube name="Vendas">
    <Table name="vendas_geral"/>
    <Dimension name="Genero" foreignKey="id_cliente">
      <Hierarchy hasAll="true" allMemberName="Todos Generos" primaryKey="id_cliente">
        <Table name="cliente"/>
        <Level name="Genero" column="genero" uniqueMembers="true"/>
      </Hierarchy>
    </Dimension>
    <Dimension name="Tempo" foreignKey="id_tempo">
      <Hierarchy hasAll="false" primaryKey="id_tempo">
        <Table name="dia"/>
        <Level name="Ano" column="ano" type="Numeric"
          uniqueMembers="true"/>
        <Level name="Trimestre" column="trimestre"
          uniqueMembers="false"/>
        <Level name="Mes" column="mes" type="Numeric"
          uniqueMembers="false"/>
      </Hierarchy>
    </Dimension>
    <Measure name="Quantidade Vendida" column="quantidade_vendida"
      aggregator="sum" formatString="#,###"/>
    <Measure name="Vendas da loja" column="vendas_loja"
      aggregator="sum" formatString="#,###.##"/>
    <CalculatedMember name="lucro" dimension="Medidas"
      formula="[Medidas].[Vendas da loja]-[Medidas].[Custos da loja]"
      <CalculatedMemberProperty name="FORMAT_STRING" value="&#,##0.00"/>
    </CalculatedMember>
  </Cube>
</Schema>
```

Figura 3: Exemplo de arquivo XML definindo um *Schema* (Adaptação de [12]).

Este *Schema* contém um único cubo, chamado “Vendas”, que possui duas dimensões, “Tempo” e “Genero”, e dois fatos, “Quantidade Vendida” e “Vendas da Loja”.

## 5 JPivot

O *Mondrian* executa as consultas e as retorna sem uma saída definida. A visualização das consultas retornadas depende de outro software, como por exemplo, o *Jpivot*. O *Jpivot* é uma *tag library*<sup>3</sup> que possibilita o acesso e disponibilização de tabelas de forma multidimensional na *Web*, além de permitir operações básicas em consultas OLAP, como *slice and dice* e *drill down*. Com o uso deste software, é possível escrever aplicativos na em JSP (*Java Server Pages*), utilizando comandos específicos e gerando saídas através de tabelas e gráficos.

O *Jpivot* é estruturado para suportar diversos servidores OLAP, especialmente *Mondrian*. Ele não utiliza a API do *Mondrian* diretamente, mas define sua própria interface OLAP. Para fazer o *Jpivot* funcionar com o *Mondrian*, estas interfaces precisam ser implementadas usando as APIs do *Mondrian*.

Os scripts escritos em JSP devem conter as consultas na linguagem MDX, *tags* relativas à sintaxe

<sup>3</sup> *Tag Library* é uma coleção de *Tags* personalizáveis que auxiliam no desenvolvimento de sistemas baseados em HTML. Alguns exemplos de tarefas que podem ser feitas por *Tag Libraries* incluem processamento de formulários, acesso à banco de dados, personalização de componentes e outros.

JSP e *tags* pertencentes à *JSP tag library* [3]. Estes scripts devem ser executados através de um navegador *Web*. O navegador submete o script a um *Web Server* (*Apache Tomcat*). O *Web Server* tratará a requisição e submeterá a consulta ao *Mondrian*, que por sua vez, submeterá ao banco de dados relacional. A consulta retornada pelo banco de dados será processada pelo *Mondrian* e os resultados remetidos à camada de apresentação, neste caso, representada pelo *Jpivot*. O *Jpivot* transforma em uma página *Web*, que é visualizada pelo mesmo navegador *Web* que fez a solicitação inicial. A Figura 4 ilustra esta arquitetura.

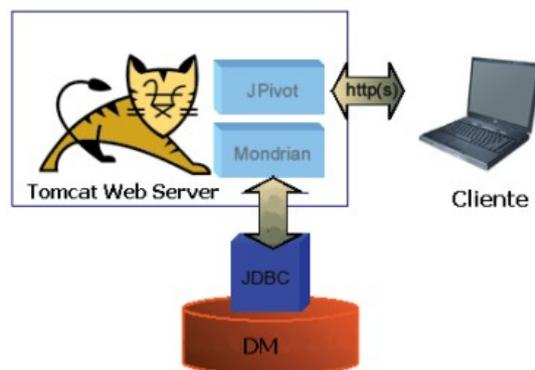


Figura 4: Arquitetura Mondrian/Jpivot

## 6 ESTUDO DE CASO

O objetivo deste trabalho é desenvolver uma solução de Data Mart e esta seção tem por finalidade descrever genericamente o funcionamento do departamento comercial de uma indústria calçadista baseado no estudo do software de gestão empresarial SFT, desenvolvido pela empresa Safetech Informática Ltda<sup>4</sup>. Baseada nas características levantadas, foi desenvolvida a aplicação para atender a necessidade deste setor.

A estrutura do departamento comercial de uma indústria calçadista é formada por representantes, gerência comercial e pessoal de apoio. Os representantes são responsáveis por realizarem as visitas aos clientes, apresentando os produtos da empresa e anotando os pedidos para repassar à fábrica. A gerência comercial é responsável pelo controle dos representantes, definição de cota de venda e acompanhamento do desempenho das vendas dos representantes e dos produtos. O pessoal de apoio é responsável pelo trabalho burocrático do setor, como atendimento a clientes, cadastramento de produtos, condições de pagamento, interação com os demais setores da empresa como produção e modelagem, entre outros.

A gerência comercial divide o país em regiões de venda. Estas regiões podem ser compostas por uma única unidade da federação (UF), mais de uma unidade em casos de UFs com pouca população ou ainda uma região pode ser somente parte de uma UF em locais com maior população. Cada região é atendida por um representante.

Cada representante recebe uma cota de venda, geralmente semanal, que o mesmo precisa cumprir para manter a fábrica com sua capacidade plena de produção. Esta cota normalmente é definida no início de cada semestre, podendo sofrer alguma alteração durante o decorrer do mesmo. O representante pode possuir prepostos, que auxiliam o mesmo no atendimento aos clientes. Cada representante recebe um percentual de comissão pelas vendas efetuadas, a qual pode ser paga pela

<sup>4</sup> Safetech Informática Ltda é uma *softwarehouse* que desenvolve sistemas de gestão empresarial com foco na indústria coureiro-calçadista.

empresa no faturamento do pedido ou na liquidação financeira do título gerado por este faturamento.

O representante faz o pedido para o cliente. O pedido contém uma determinada data de entrega, uma condição de pagamento, a transportadora que fará entrega e os itens do pedido, que são os produtos comprados pelo cliente. O pedido pode ser de dois tipos: produção ou pronta entrega. Nos pedidos de produção, a fábrica recebe o pedido, produz, fatura e entrega ao cliente. Já no pedido de pronta entrega, os produtos estão prontos e disponíveis para entrega. Neste caso, os produtos podem ser fabricados para este fim ou serem oriundos de cancelamento de algum pedido de produção que já se encontrava em processo de fabricação, mas que por algum motivo o cliente não deseja mais receber a mercadoria. Nesta situação, a empresa então disponibiliza este produto para ser vendido como pronta entrega.

As empresas criam coleções de produtos. No caso de calçados femininos, por exemplo, geralmente são criadas duas coleções por ano, uma a cada início de semestre, sendo classificadas normalmente como primavera-verão e outono-inverno. Cada coleção é composta por várias linhas de produtos. Cada linha é composta por vários modelos que possuem características em comum. Estes modelos podem ser produzidos em mais de um material e em uma variada combinação de cores.

Outro processo que precisa ser acompanhado pelo departamento comercial, é a devolução de venda. A devolução é solicitada pelo cliente quando este, por algum motivo, devolve a mercadoria para fábrica. A fábrica avaliará o produto e o motivo alegado pelo cliente para tomar a devida providência, que poderá ser a reposição do produto ou a indenização do cliente mediante compensação financeira.

A gerência comercial necessita fazer o acompanhamento constante do desempenho da área para identificar comportamentos e agir proativamente. Para tanto, precisa analisar as informações sob vários ângulos e diferentes cruzamentos. Esta é uma das principais dificuldades enfrentadas pelos desenvolvedores de soluções para esta área, que utilizam ferramentas tradicionais para criar relatórios estáticos. Estas ferramentas somente apresentam a informação em um formato pré-determinado, não permitindo ao usuário uma maior interatividade. Além disto, é necessária a criação de um número muito grande de relatórios nos mais variados formatos para atender às necessidades dos usuários, e estes, em diversas situações, precisam emitir vários relatórios para finalmente chegar à informação desejada.

Com base neste cenário, foi construído um Data Mart, composto pelas tabelas de fatos e dimensões exibidas no modelo dimensional ilustrado na Figura 5.

A ferramenta OLAP foi desenvolvida seguindo-se a arquitetura de DM independente, ou seja, inicialmente será um DM sem qualquer integração com outros DMs ou DWs que a empresa possa utilizar. Este DM utiliza a arquitetura de armazenamento ROLAP, isto é, os dados estão armazenados em um banco de dados relacional. A camada OLAP é desempenhada pelo software *Mondrian*, que faz o acesso a esta base de dados através do envio de consultas na linguagem MDX. Ao receber o resultado da consulta, o *Mondrian* monta o cubo de dados, que por sua vez, é acessado pelo *Jpivot*, camada responsável pela apresentação dos dados ao usuário. Esta solução foi desenvolvida para a arquitetura *Web*, e é disponibilizada através do servidor *Apache Tomcat*.

Os dados do DM vêm do módulo comercial do sistema de gestão SFT, e por isto a ferramenta desenvolvida foi denominada de *SFT Analysis* e complementarà a solução disponibilizada pela empresa Safetech Informática LTDA, visando no futuro, ser um módulo que esta possa oferecer a seus clientes.

O DM foi criado no mesmo servidor de banco de dados do sistema SFT. O banco de dados utilizado é o *Oracle 9i*. Porém este não é um requisito obrigatório do sistema, pois como o banco de dados é acessado pelo *Mondrian* através de *JDBC*, pode ser utilizado qualquer banco de dados compatível com este padrão, incluindo os bancos de dados *freeware* mais conhecidos como o *Mysql* e o *Postgresql*.

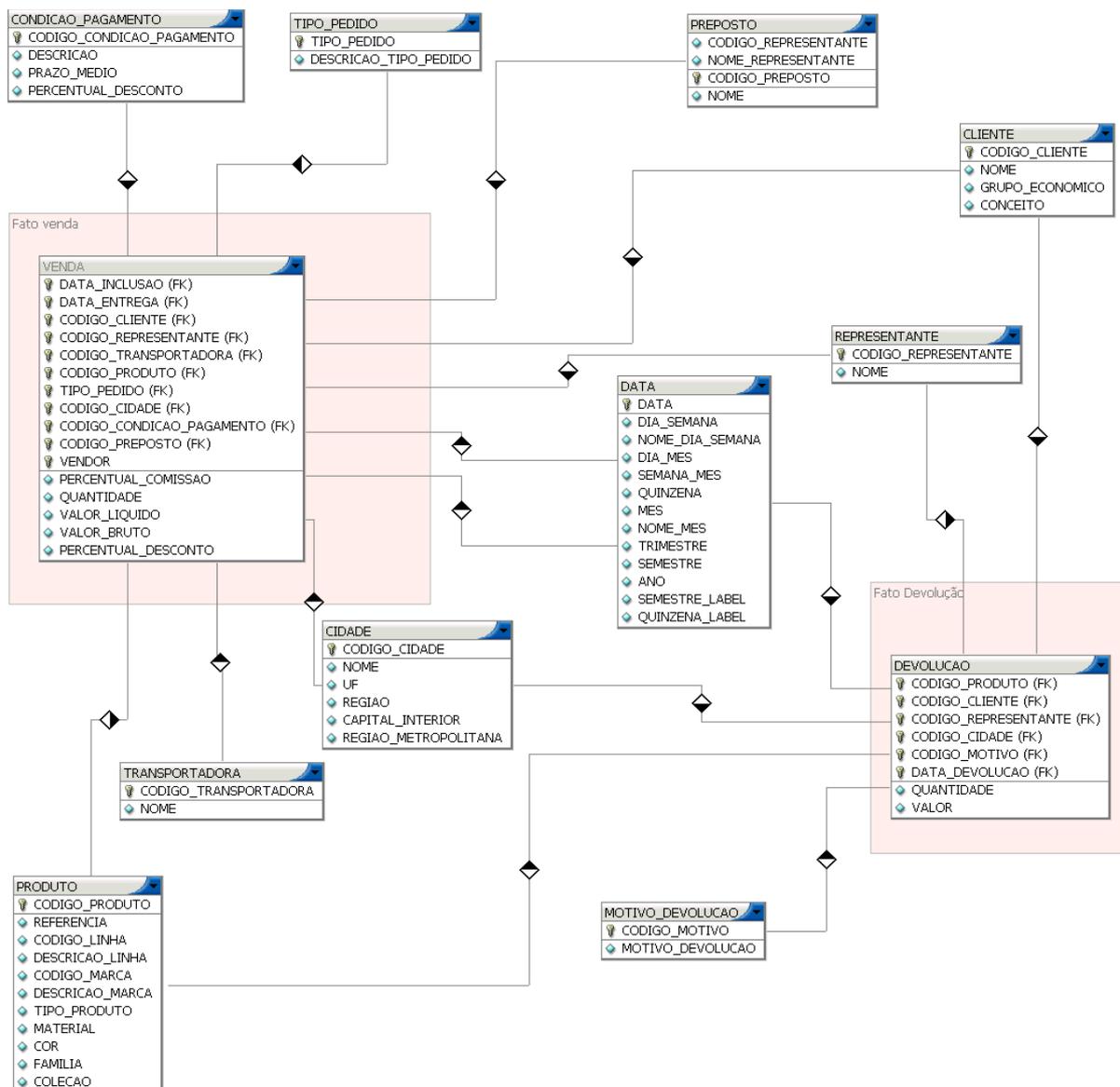


Figura 5: Modelo dimensional da ferramenta OLAP SFT Analysis

A etapa de extração, transformação e carga do DM é realizada através da execução de *scripts* SQLs que fazem a leitura dos dados na base do sistema SFT os inserem no DM. A carga dos dados no DM é sempre uma carga total, ou seja, os dados existentes são eliminados, e após este procedimento, todo o DM é populado novamente. Optou-se por este procedimento porque o volume

de dados não é muito extenso e durante as simulações desta carga o tempo dispensado nesta tarefa foi bastante baixo, inferior a uma hora de duração, o que permite que esta carga possa ser programada para rodar diariamente, fora de horário de utilização do sistema transacional da empresa, com absoluta tranqüilidade.

Na figura 6 pode-se observar, em sua parte superior, um conjunto de botões que permitem ao usuário, executar diversas operações dimensionais sobre um determinado cubo. Estes botões fazem parte da *tag library Jpivot*, que pode ser estendida, através de *JSP/Java*. Neste trabalho, foram adicionados dois botões, um deles com o objetivo de salvar consultas e o outro para excluí-las.

A escolha de um dos três cubos para ser analisado é realizada no menu existente na parte esquerda da ferramenta. Escolhido o cubo, este pode ser navegado, gerando dinamicamente o resultado das consultas. Gráficos são adicionados à consulta com simples acesso a barra de botões. As consultas geradas podem ser salvas, exportadas para formatos como PDF e CSV. Além disso, as medidas e atributos das tabelas também podem ser exibidos e ocultados através da barra de ferramentas.

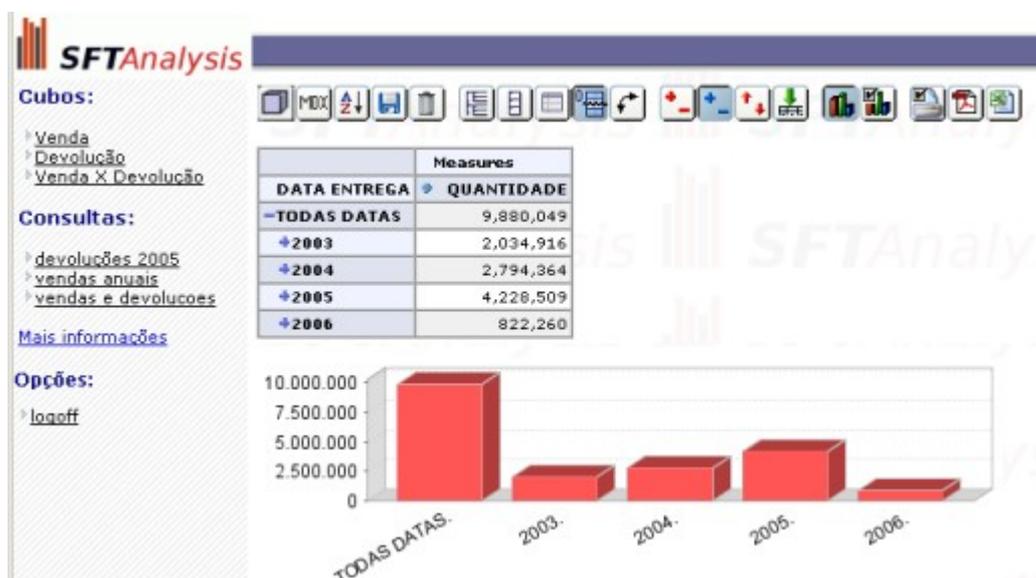


Figura 6: Ferramenta OLAP SFT Analysis

## 7 CONCLUSÃO E TRABALHOS FUTUROS

Com o desenvolvimento da aplicação, pôde-se comprovar a viabilidade da utilização de ferramentas gratuitas e/ou *Open Source* para criar SADs, pois foi possível criar um DM e uma ferramenta OLAP utilizando-se exclusivamente estas ferramentas, e elas apresentaram, além de uma boa facilidade de utilização, uma diversidade de recursos que oferecem excelentes opções de análises de informações. Um dos principais pontos a salientar no uso destas ferramentas é a possibilidade de adição de novos recursos, ampliando a gama de funcionalidades, tornando a sua utilização mais atraente para os desenvolvedores de soluções de BI. O trabalho em questão permitiu a aprendizagem de como estender o código da *tag library Jpivot*, incluir *Jpivot* e *Mondrian* em uma aplicação, a criação de cubos multidimensionais, a aplicação de filtros e operações OLAP sobre estes cubos, dentre outros temas relacionados às ferramentas utilizadas.

Um dos pontos a serem melhorados nestas ferramentas é a criação de uma documentação mais abrangente para permitir que os desenvolvedores possam extrair o máximo de recursos das mesmas. Abre-se também a oportunidade de trabalhos futuros na área, como a adição de novos recursos à ferramenta desenvolvida e a criação de novos DMs em outros setores da indústria calçadista.

## REFERÊNCIAS

- [1] Barbalho, Patrícia. Descubra o *Data Warehouse*: produtividade e rapidez. Revista SQL magazine. Edição 3, 2003.
- [2] Barbieri, Carlos. *Business Intelligence*: modelagem e tecnologia. Rio de Janeiro: Axcel Books, 2001.
- [3] Brito, Maiquel de. Proposta de um *Data Warehouse* de informações acadêmicas. Novo Hamburgo: 2004. 111 p. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo.
- [4] Campos, Maria Luiza; Filho, Arnaldo V. Rocha. *Data Warehouse*. Disponível em <<http://genesis.nce.ufrj.br/dataware/tutorial/home.html>>. Acesso em: 20 de set. 2005.
- [5] Grimes, Seth. *Open-Source Releases Invade the Reporting Market*. *Intelligent Enterprise*. Estados Unidos, 2005. Disponível em: <<http://www.iemagazine.com/showArticle.jhtml?articleID=163100786>>. Acesso em: 06 ago. 2005.
- [6] Hokama, Daniele Del Bianco et al. A modelagem de dados no ambiente *Data Warehouse*. São Paulo: 2004. 121 p. Projeto de Diplomação (Bacharelado em Sistemas de Informação) – Faculdade de Computação e Informática, Universidade Presbiteriana Mackenzie. Disponível em: <<http://meusite.mackenzie.com.br/rogerio/tgi/2004ModelagemDW.pdf>>. Acesso em: 20 set. 2005.
- [7] Inmon, W. H. Como construir o *Data Warehouse*. 2. ed. Rio de Janeiro: Campus, 1997.
- [8] Inmon, W. H. et al. Gerenciando *Data Warehouse*. São Paulo: Makron Books, 1999.
- [9] Kimball, Ralph; Merz, Richard. *Data Webhouse*: Construindo o *Data Warehouse* para a Web. Tradução: Edson Furmankiewicz, Joana Figueiredo. 1. ed. Rio de Janeiro: Campus, 2000.
- [10] Kimball, Ralph; Ross, Margy. *The Data Warehouse Toolkit*: guia completo para modelagem dimensional. 2. ed. Rio de Janeiro: Campus, 2002.
- [11] Microsoft. *Microsoft Corporation*. Disponível em: <<http://www.microsoft.com>>. Acesso em: 06 ago. 2005.
- [12] Mondrian. *Mondrian OLAP Server*. Disponível em: <<http://mondrian.sourceforge.net>>. Acesso em: 06 ago. 2005.
- [13] Pernas, Ana Marilza da Rosa. Modelagem de um *Data Webhouse* voltado a produção e comercialização de sementes. Pelotas: 2003. 70 p. Projeto de Diplomação (Bacharelado em Ciência da Computação) – Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas. Disponível em: <[http://www.ufpel.edu.br/prg/sisbi/bibct/acervo/info/2003/mono\\_ana\\_pernas.pdf](http://www.ufpel.edu.br/prg/sisbi/bibct/acervo/info/2003/mono_ana_pernas.pdf)>. Acesso em: 20 set. 2005.
- [14] Singh, Harry. *Data Warehouse*. São Paulo: Makron Books, 2001.
- [15] Thomsen, Erik. OLAP: construindo sistemas de informações multidimensionais. 2. ed. Rio de Janeiro: Campus, 2002.