# Projeto de Data Mart utilizando ferramentas Open Source

**Braian O. Dias**[1]

[1]Curso de Ciência da Computação– Universidade de Passo Fundo (UPF)
99001-970 – Passo Fundo – RS – Brasil

[1]braian.d@gmail.com

***Resumo.*** *Atualmente, as técnicas de Business Intelligence auxiliam na tomada de decisão, tornando-se fatores decisivos para o futuro de uma organização. Neste contexto, o presente trabalho visa apresentar o projeto e a implementação de uma das técnicas bastante difundidas deste assunto, o Data Warehouse. Para isso será criado um Data Mart com auxílio de ferramentas de código aberto com o intuito de, posteriormente, servir como apoio para o processo de tomada de decisão dentro de uma empresa.*

## 1. Introdução

Durante anos o desenvolvimento de sistemas de informação tinha como foco principal o ambiente operacional, com direcionamento na automação dos processos, obtenção de dados de forma confiável, com nível mínimo de erro. A partir da década de 1980 começou a se cunhar termo *Business Intelligence* (BI), ou inteligência do negócio, que pode ser definido como o conjunto de tecnologias orientadas a disponibilizar informação e conhecimento em uma empresa . O uso de BI provê visões históricas, atuais e futuras sobre operações de negócios, geralmente usando dados armazenados em um *Data Warehouse* (DW) ou um *Data Mart* (DM).

*Data Warehouse* são bases de dados históricas voltadas a um ou mais assuntos específicos de negócios. Essas bases de dados refletem a história da empresa em um determinado período, permitindo a análise de dados e descoberta de informações antes inimagináveis, ou que despediam muito tempo para encontrá-las, com a capacidade de disponibilizar a visualização dos dados sob diversas perspectivas e de navegar no nível de detalhe da informação. Por definição os dados de um *Data Warehouse* não são voláteis, ou seja, não mudam com o passar do tempo. Estes dados são obtidos através de um processo chamado de Extração, Transformação e Carga sobre as bases de dados dos sistemas transacionais. Após os dados estarem carregados no *Data Warehouse* os mesmos estarão disponíveis apenas para leitura, e a atualização desta base de dados se dará de forma incremental.

O *Data Mart*, que é definido como um subconjunto do *Data Warehouse* focado em um assunto específico, usa um modelo de dados multidimensional, também chamado de cubo, que possui algumas diferenças em comparação com o modelo relacional tradicional. A modelagem multidimensional é uma técnica de concepção e visualização de um modelo de dados de um conjunto de medidas que descrevem aspectos comuns de negócio. Sua utilização ajuda na sumarização e reestruturação dos dados e apresenta visões que suportam a análise dos valores destes dados.

Para a visualização da informação contida nos *Data Marts* surgiram as ferramentas OLAP (*On Line Analytical Processing*, ou Processo Analítico em Tempo Real), que permitem a visualização dos dados sobre diferentes níveis de detalhamento

de um cubo de dados. Através das operações de *Drill* é possível aumentar (*Drill down*) ou diminuir (*Drill up*) o nível de detalhamento dos dados. As operações de *Drill* bem como outras operações utilizadas em ferramentas OLAP se apóiam na capacidade de manipular dados e formas de apresentação de maneira rápida que a modelagem multidimensional provê. Com o uso destes recursos o usuário pode então navegar através das informações de maneira rápida e flexível.

Tendo como base a necessidade das organizações em ter informações para análise e tomada de decisão, o presente trabalho tem como objetivo apresentar os processos envolvidos na construção de um sistema que utiliza-se da tecnologia de *Data Warehouse*. Os principais conceitos relacionados à área serão apresentados na seção 2. Na seção 3 será apresentada a metodologia utilizada para a criação do *Data Warehouse*. A seção 4 mostra a aplicação do trabalho em uma empresa da região. Por fim, a seção 5 apresenta as conclusões acerca do artigo.

## 2. Data Warehouse

A história do *Data warehouse* inicia com a evolução dos sistemas de apoio a decisão (SAD), com o objetivo de criar um repositório central de dados históricos para consultas e análises (INMON,2005).

Nos anos 60 os sistemas de informação eram construídos em linguagens como *Fortran* e *COBOL*, e usavam fitas magnéticas como meio de armazenamento, dificultando o acesso aos dados (o acesso se dava da forma seqüencial). O acesso nesse meio era lento tornando praticamente impossível o uso para análises rápidas. Era comum também a utilização de várias fitas magnéticas, criando muitas vezes redundância de dados.

Na década de 70 surgiu o DASD (*direct access storage device*, dispositivo de armazenamento de acesso direto, ou simplesmente armazenamento em disco), aumentando relativamente o desempenho ao acesso dos dados por se dar de forma direta, e não seqüencial como nas fitas magnéticas. Em conjunto com esse advento surgiu um tipo de software chamado de SGBD (sistema gerenciador de banco de dados). O propósito do SGBD era tornar fácil para o programador armazenar e acessar dados. Com isto também surgiu o conceito de base de dados, que diferente do armazenamento em fita, era uma única fonte de dados para todo o processamento. Na metade dos anos 70 surgiram os sistemas OLTP (*Online Transaction Processing*) tornando o acesso rápido aos dados possível, abrindo assim novas perspectivas de negócios como sistemas de reservas, controle de manufatura e assim por diante.

Nos anos 80 surgiram novas tecnologias como o PC e as linguagens de programação de quarta geração. Com essas tecnologias o usuário final começou a assumir um novo papel, o de controlar dados e sistemas, papel este que antes era restrito ao domínio do processamento de dados. Com essas novas possibilidades começou a ser percebido que poderia ser feito mais com dados do que simplesmente processamento de transações em tempo real. Surgiram os *Management Information Systems* (MIS), ou sistemas de informações gerenciais, hoje conhecidos como Sistemas de apoio a decisão (SAD). Estes sistemas eram usados para guiar decisões gerenciais, mudando o paradigma anterior onde dados eram usados estritamente para conceber decisões operacionais. Porém o processamento de transações online e o processamento analítico ainda eram feitos simultaneamente sobre a mesma base de dados.

Com a necessidade de informações mais detalhadas os primeiros programas para extração de dados começaram a surgir. Estes programas buscavam dados que eram relevantes e os transportavam para outra base de dados separada dos sistemas transacionais. Esta tarefa se tornou comum para criar relatórios de análise, onde um determinado departamento realizava a extração dos dados pré-definidos com o intuito de gerar informações para os níveis mais altos da administração. Porém juntamente com estes programas surgiram alguns problemas inerentes ao fato da não estruturação dos processos de extração, tais como a inconsistência de dados, dados sem base no tempo, diferença de algoritmos de busca.

Estes processos de extrações sem controle começaram a formar uma espécie de "teia de aranha", a qual é definida como "arquitetura de desenvolvimento espontâneo" (INMON,2005). Cada programa de extração gerava outra base de dados. Depois eram realizadas extrações das extrações, e isso continuava sem qualquer controle. Esta arquitetura gerou muitos problemas, sobretudo a falta de credibilidade dos dados, baixa produtividade do processo, e a impossibilidade de transformar dados em informações.

Os problemas encontrados nesta arquitetura de desenvolvimento espontâneo levaram a criação de um ambiente projetado que satisfizesse as necessidades dos SAD, o ambiente do *Data Warehouse*. Este ambiente mudou a percepção sobre os dados, diferenciando dados operacionais e dados para apoio a decisão. A figura 1 mostra as principais diferenças entre os dados.

| DADOS PRIMITIVOS/DADOS OPERACIONAIS | DADOS DERIVADOS/DADOS SAD |
|---|---|
| • baseados em aplicações | • baseados em assuntos ou negócios |
| • detalhados | • resumidos, ou refinados |
| • exatos em relação ao momento do acesso | • representam valores de momentos já decorridos ou instantâneos |
| • atendem à comunidade funcional | • atendem à comunidade gerencial |
| • podem ser atualizados | • não são atualizados |
| • são processados repetitivamente | • processados de forma heurística |
| • requisitos de processamento conhecidos com antecedência | • requisitos de processamento não são conhecidos com antecedência |
| • compatíveis com o SDLC | • ciclo de vida completamente diferente |
| • a performance é fundamental | • performance atenuada |
| • acessados uma unidade por vez | • acessados um conjunto por vez |
| • voltados para transações | • voltados para análise |
| • o controle de atualizações é atribuição de quem tem a posse | • o controle de atualizações não é problema |
| • alta disponibilidade | • disponibilidade atenuada |
| • gerenciados em sua totalidade | • gerenciados por subconjuntos |
| • não contemplam a redundância | • a redundância não pode ser ignorada |
| • estrutura fixa; conteúdos variáveis | • estrutura flexível |
| • pequena quantidade de dados usada em um processo | • grande quantidade de dados usada em um processo |
| • atendem às necessidades cotidianas | • atendem às necessidades gerenciais |
| • alta probabilidade de acesso | • baixa, ou modesta probabilidade de acesso |

Fonte: Inmon William H., Como Construir o Data Warehouse, página 18.

**Figura 1. Diferença entre dados primitivos e derivados.**

Com a separação dos dados entre primitivos e derivados surgiram os níveis da arquitetura projetada para *Data Warehouse*: Nível operacional, *Data Warehouse*, departamental e individual. No nível operacional são encontrados apenas dados oriundos dos sistemas de processamento de transações, ou seja, dados puramente primitivos. O nível do *Data Warehouse* traz dados primitivos que não tem atualização

freqüente e também dados derivados. No nível departamental são encontrados quase que apenas dados derivados e é no nível individual que a maioria das análises heurísticas é realizada. O nível individual geralmente é temporário e de pequenas proporções por ser utilizado apenas para processar informações necessárias em um determinado período.

Um *Data Warehouse* é um repositório central de dados, históricos e atuais, que podem representar algum valor para os tomadores de decisão dentro de uma organização. Ele consolida e padroniza as informações oriundas das diversas fontes de dados da empresa, tornando-as disponíveis para análise gerencial e tomada de decisões por toda a empresa (LAUDON,2007).

O *Data Warehouse* requer uma arquitetura onde se possa visualizar inicialmente a partir do todo e então descendo aos níveis de detalhe. Detalhes que em um contexto mais amplo se tornam importantes.

Na arquitetura projetada existe o *Data Mart*, que segundo Laudon (2007) "é uma subconjunto de um *Data Warehouse*, no qual uma porção resumida ou altamente focalizada dos dados da organização é colocada em um banco separado destinado a uma população específica de usuários".

Conforme Inmon (2005), um *Data Mart* é uma estrutura de dados dedicada em servir as necessidades de análise de um grupo de pessoas, tais como departamento de finanças ou recursos humanos.

### 2.1. Características do *Data Warehouse*

Um *Data Warehouse* integra e consolida informações de fontes internas e externas, sumarizando, filtrando e limpando esses dados, preparando-os para análise e suporte à decisão (MACHADO,2000). A tecnologia de DW possui um conjunto de características que a distingue dos outros sistemas convencionais:

- Extração de dados de fontes heterogêneas (internas ou externas);
- Transformação e integração dos dados antes de sua carga;
- Visualização dos dados em diferentes níveis;
- Utilização de ferramentas voltadas para acesso com diferentes níveis de apresentação;
- Dados somente são inseridos, sem a possibilidade de atualização.

Segundo Inmon (2005), um *Data Warehouse* é orientado a assuntos, integrado, não volátil, e variável em relação ao tempo.

Um *Data Warehouse* é orientado a assunto, diferente dos sistemas transacionais existentes que são orientados por processos comuns. Em uma companhia de seguros, por exemplo, as aplicações transacionais poderiam estar divididas em automóvel, vida, saúde, e perdas. Já as áreas de assunto para essa companhia poderiam estar divididas em cliente, apólice, prêmio e indenização.

Outra característica de suma importância é integração dos dados. Isto quer dizer que as inconsistências encontradas em dados provenientes de diversos sistemas são desfeitas. Um exemplo é o caso da codificação do gênero, onde podem existir diversas

formas de representação entre os sistemas, porém, ao serem incorporados ao *Data Warehouse*, estes dados necessitam representar o mesmo fato, independente da sua codificação inicial.

A terceira característica consiste em que o *Data Warehouse* não é volátil. No ambiente transacional os dados são inseridos, normalmente, um por vez, e podem sofrer atualizações ao longo do tempo. Em contrapartida os dados contidos no *Data Warehouse* são carregados e acessados normalmente em grandes quantidades, e a atualização dos dados (geralmente) não ocorre neste ambiente. As operações de processamento neste ambiente se restringem a duas: inclusão de novos registros e consulta aos registros existentes.

A última característica diz respeito ao fato de ele ser variável em relação ao tempo. Isto quer dizer que os dados do *Data Warehouse* representam resultados operacionais em um determinado momento de tempo, o momento em que foram capturados da base de dados operacional. Como estes dados representam um estado da organização em um período de tempo eles não podem ser atualizados. Por exemplo, o dado relativo a vendas de um determinado mês nunca mais terá seu valor modificado. No ambiente operacional os dados representam o valor corrente de alguma coisa, é são validos para determinado instante, podendo ser alterados. Na estrutura do *Data Warehouse* sempre haverá um elemento representando tempo, indicando a validade de um determinado dado. Outro ponto importante com relação ao tempo é que os espaços de tempo usados para análise são significativamente maiores se comparados com o ambiente transacional. Um horizonte de 60 a 90 dias pode ser satisfatório se tratando de um ambiente transacional. Já em um *Data warehouse* um horizonte de tempo de 5 a 10 anos é considerado normal.

### 2.2. Granularidade

Inmon (2005) descreve que, “A granularidade diz respeito ao nível de detalhe ou de resumo contido nas unidade de dados existentes no *Data Warehouse*”. Isto quer dizer que quanto mais detalhado for o dado, mais baixa será a granularidade. Logo, quanto menos detalhado for, mais alto será o nível de granularidade. A escolha da granularidade reflete principalmente no volume de dados contido no *Data Warehouse* e, conseqüentemente , na performance dele. O nível de detalhe escolhido depende da necessidade da informação a ser obtida. Quando definido um nível de detalhe muito baixo, é necessário observar que estes dados não poderão ser decompostos futuramente para uma análise com um nível maior de detalhes. Entretanto, com uma granularidade baixa, é possível agrupar as informações para que sejam visualizadas com um nível menor de detalhes, mais sumarizados.

### 2.3. Modelos de dados multidimensionais

A modelagem multidimensional, segundo Machado (2000), “é uma técnica de concepção e visualização de um modelo de dados de um conjunto de medidas que descrevem aspectos comuns de negócios”. Ela é utilizada para prover uma visão que suporte a análise de informações de forma flexível e rápida. Em um modelo multidimensional são encontrados três elementos básicos: fatos, dimensões e medidas.

Um fato representa um item, transação ou evento de um negócio. É tudo aquilo que reflete a evolução dos negócios de uma organização, o que é mensurável. Estes

fatos possuem valores numéricos, denominadas medidas, que identificam um fato ao longo do tempo. As medidas servem como um indicador de negócios relativo às dimensões que participam deste fato. Em nível de implementação, um fato é representado por uma "tabela de fato", que possui atributos numéricos que servem como indicadores, as medidas, e as chaves que ligam os fatos as dimensões que o compõem.

Dimensões são elementos que participam de um assunto de negócios, são os pontos de vista pelos quais serão feitas as análises dos fatos. Cada dimensão faz parte de um fato e, normalmente, não existem valores quantitativos dentro das tabelas das dimensões, apenas atributos descritivos (texto), devido ao fato que elas são usadas para a classificação dos fatos. Em uma tabela de dimensão normalmente serão encontrados um menor numero de registros se comparado com a tabela de fatos.

Cada dimensão pode conter vários membros que são usados para classificação de dados dentro de uma dimensão específica. Estes membros são os atributos das tabelas de dimensão no projeto físico. Os membros por sua vez são organizados de forma hierárquica para que possam ser agregados de acordo com seus níveis. Em um exemplo de dimensão de tempo, uma estrutura hierárquica poderia ser equivalente a apresentada no quadro 1.

| |
|---|
| Ano |
| Trimestre |
| Mês |

**Quadro 1. Hierarquia da dimensão Tempo.**

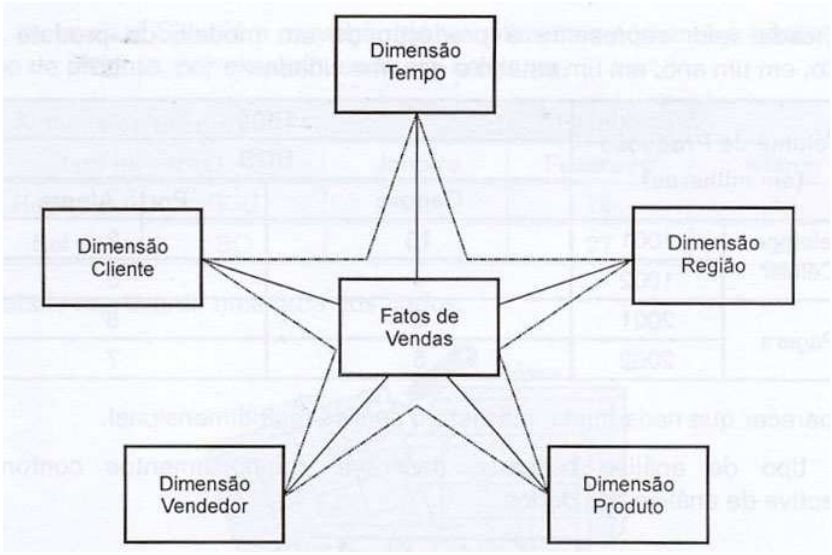
Nesta hierarquia é possível observar que os valores poderão ser agregados mensalmente, trimestralmente e anualmente. Os níveis em que se apresentam cada um dos membros da dimensão tempo estão assim dispostos devido ao fato que um trimestre é um conjunto de meses, e um ano é um conjunto de trimestres. Logo os dados poderão ser agregados futuramente de acordo com diferentes níveis de hierarquia.

O modelo multidimensional composto pelos elementos apresentados é usualmente representado por um cubo, em uma analogia a visualização de informações sob a perspectiva de três dimensões. Entretanto um modelo multidimensional pode comportar mais de três dimensões, o chamado *hipercubo*. A visualização teórica deste tipo de estrutura é muito complicada, por isso se usa o termo cubo em referência a qualquer modelo de dados multidimensional.

### 2.3.1. Modelo *Star* ou Estrela

O modelo estrela é um modelo que implementa os conceitos de estrutura multidimensional onde, na sua composição típica, apresenta uma grande entidade central denominada fato e um conjunto de entidades menores denominadas dimensões, arranjadas ao redor da entidade fato, de forma que a representação gráfica do modelo se pareça com uma estrela. Contrapondo um modelo ER tradicional, onde é possível ter várias formas de diagramação, no modelo multidimensional o modelo se apresenta praticamente da mesma forma. As dimensões neste modelo apresentam certa redundância nos dados, devido a desnormalização. O relacionamento da entidade fato no modelo com as entidades dimensões é uma simples ligação de um para muitos no

sentido da dimensão para o fato. Comparado com o modelo ER tradicional é possível observar que a entidade fato se trata de uma entidade associativa.

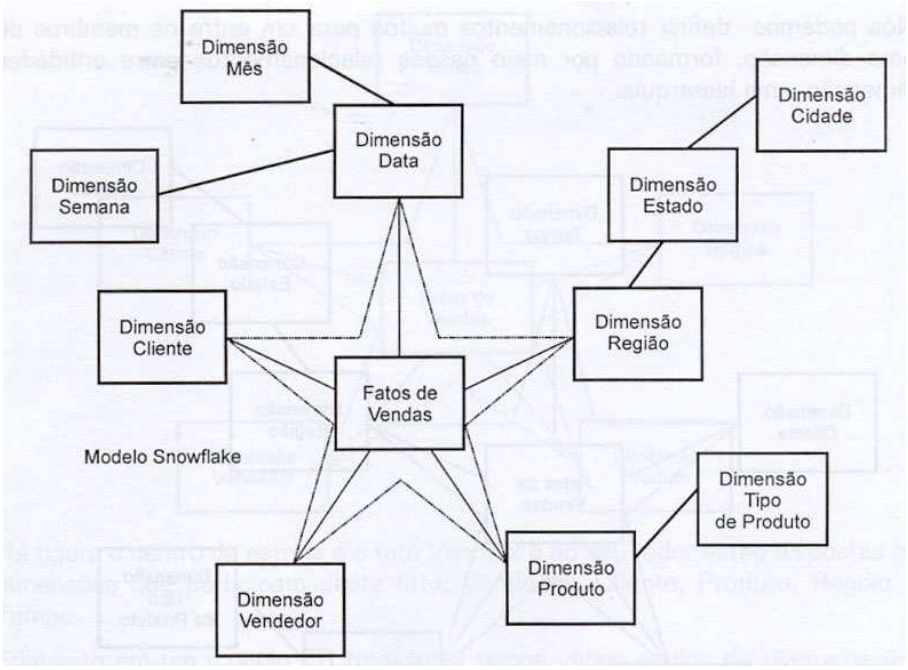



Fonte: Machado, “Projeto de Data Warehouse: uma visão multidimensional”, página 74.

**Figura 2. Modelo Estrela.**

### 2.3.2. Modelo *SnowFlake*

O modelo *SnowFlake* apresenta a mesma estrutura do modelo estrela, com uma tabela de fato no centro e as dimensões em sua volta, porém os dados das dimensões possuem hierarquias entre seus membros. Devido ao fato de estar normalizado, o modelo evita redundância de dados nas tabelas dimensão.




Fonte: Machado, “Projeto de Data Warehouse: uma visão multidimensional”, página 76.

**Figura 3. Modelo SnowFlake.**

## 2.4. ETL

Extract/transform/load – ETL, ou extração, transformação e carga, é o processo onde serão captados os dados das diversas fontes existentes na organização para popular o *Data Warehouse*. A extração consiste na obtenção destes dados, das diversas fontes disponíveis, sejam elas bases de dados, arquivo textos ou binários, páginas da *web*. Os dados obtidos são então transformados do ambiente operacional para o ambiente do *Data Warehouse*, onde os dados são integrados, refinados, transformados e sumarizados. No processo de carga o volume de dados processado é levado ao *Data Warehouse*, onde estará disponível para análise.

## 2.5 Ferramentas OLAP

No ambiente departamental, ou ambiente do *Data Mart*, existem ferramentas do tipo OLAP que permitem acesso as informações de forma rápida, flexível e intuitiva, facilitando a visualização dos dados do cubo.

Existem algumas operações básicas para este tipo de ferramenta, que são usadas para a análise dos dados : *Drill down*, *Roll up*, *Slice* e *Dice*.

*Drill down* e *Roll up* são operações realizadas dentro dos níveis hierárquicos de cada dimensão. Um *Drill down* aumenta o nível de detalhamento da informação, ou seja, parte de um nível menos detalhado para um mais detalhado. *Roll up* é a operação inversa, o nível de detalhe é diminuído. Por exemplo, em uma dimensão de tempo a operação de *Drill down* detalharia as vendas apresentadas anteriormente em ano por trimestres. Diminuindo o nível de detalhes do nível de meses para trimestres seria um exemplo de operação *Roll up*.

*Slice* e *Dice* são operações usadas para navegação das informações do cubo. *Slice* é um corte no cubo que mantêm a mesma perspectiva na visualização dos dados, é apresentada uma fatia dos dados. No caso da visualização de alguns tipos de clientes, é possível, por exemplo, apresentar somente aquelas do sexo feminino. A operação *Dice*, representada na figura 4, é uma mudança na perspectiva da visualização no cubo.

| Volume de Produção (em milhares) | | 1999 | | | |
|---|---|---|---|---|---|
| | | Telefone Celular | | Pagers | |
| | | 1001 | 1002 | 2001 | 2002 |
| RGS | Canoas | 13 | 4 | 2 | 5 |
| | Porto Alegre | 20 | 8 | 6 | 7 |

Dice

| Volume de Produção (em milhares) | | 1999 | |
|---|---|---|---|
| | | RGS | |
| | | Canoas | Porto Alegre |
| Telefone Celular | 1001 | 13 | 2 |
| | 1002 | 4 | 5 |
| Pagers | 2001 | 2 | 6 |
| | 2002 | 5 | 7 |

Fonte: Machado, “Projeto de Data Warehouse: uma visão multidimensional”, páginas 72 e 73.

**Figura 4. Exemplo de uma operação Dice sobre um cubo.**

Em um primeiro momento esta sendo analisado o volume produzido em um estado, em uma cidade, em um ano, de um modelo de produto e de um produto. Com a mudança de perspectiva os valores agora representam a produção de um modelo de produto e um produto, em um ano, em um estado e em uma cidade. Estas operações permitem descobrir comportamentos conforme a perspectiva de análise dos dados.

## 3. Metodologia

Conforme apresentado, um *Data Warehouse* é composto por repositórios centrais, voltados a assuntos ou área específica de negócios, chamados de *Data Marts*.

Machado (2000) apresenta três arquiteturas de *Data Warehouse*, sendo que a escolha da arquitetura normalmente é baseada no ambiente de negócios, ou porte da empresa, e nos recursos disponíveis. As arquiteturas apresentadas são: Global, independente e integrada. Já para a implementação, existem as abordagens *top-down* , *bottom-up* e a combinação das duas.

### 3.1. Arquitetura do DW

Na arquitetura global o *Data Warehouse* é construído visando o armazenamento de informações da empresa como um todo. Ele é considerado um repositório único com informações disponíveis para toda a empresa.

Na independente, são criados *Data Marts* que atendem à necessidades específicas e departamentais, sem foco na empresa como um todo. Estes por sua vez não possuem conectividade com outros *Data Marts* voltados a outros assuntos.

A arquitetura de *Data Marts* integrados traz dados departamentais separados em cada um dos *Data Marts*, porém eles são integrados, ou interligados de alguma forma. Os *Data Marts* podem compartilhar informações entre si, tornando possível uma visão mais globalizada das informações da empresa.

### 3.2. Tipos de implementação

A abordagem de implementação *top-down* requer que seja feito um maior planejamento e definições no inicio do projeto do *Data Warehouse*. Isto requer a análise de todas as fontes de dados existentes, em todos os departamentos, para que ele seja projetado levando em consideração a empresa como um todo. O resultado dessa abordagem é um *Data Warehouse* com abrangência global da empresa no momento de sua implementação. Somente após o termino do projeto do *Data Warehouse* que será possível definir os *Data Marts*, onde serão feitas as consultas efetivamente.

A implementação *bottom-up* permite que sejam construídos *Data Marts* sem ter uma visão global do *Data Warehouse* da empresa. A criação de *Data Marts* de forma incremental faz com que a visão global da empresa seja atingida paulatinamente. Esta abordagem geralmente inicia com a criação de um DM específico, seguindo por diversos outros que, no final, se transformarão no *Data Warehouse*.

Na implementação combinada, ou seja, uma implementação integrando a abordagem *top-down* e *bottom-up* ao mesmo tempo, é realizada uma modelagem do *Data Warehouse* de forma macro, sendo que a partir deste modelo serão escolhidas áreas de interesse para implementação dos *Data Marts*. Cada *Data Mart* gerado terá como base o modelo inicial criado para o *Data Warehouse*.

## 4. Trabalho Realizado

Após o estudo teórico dos conceitos apresentados na literatura foi iniciada uma busca de ferramentas e métodos para criação de um *Data Warehouse* e de ferramentas para acesso das informações obtidas. Antes disso, porém, realizou-se o estudo do ambiente de implementação, com o intuito de encontrar as ferramentas e métodos adequados para este ambiente específico.

### 4.1. Exemplo de aplicação

A aplicação deste trabalho se deu em uma empresa do setor de comércio varejista que atualmente dispõe de diversos sistemas transacionais e gerenciais, os quais alimentam uma base de dados única para todos esses sistemas. Segundo dados da própria empresa, estes sistemas começaram a ser utilizados no ano de 2005, com foco na automatização e controle das atividades desenvolvidas. Com o passar dos anos o volume de dados cresceu, e hoje a base de dados possui um tamanho de aproximadamente 4,5 *gigabytes*, sendo que a tabela onde são armazenados os dados das vendas, item a item, possui cerca de 11.773.612 (onze milhões, setecentos e setenta e três mil, seiscentos e doze) registros.

Para a execução dos processos de ETL utilizou-se um computador *Pentium Dual Core* com processador de dois núcleos, 1 GB de memória *RAM*, HD de 160 GB e sistema operacional *Windows XP*. O banco de dados operacional da empresa fica armazenado em um servidor dedicado para isso, portanto, para o processo de ETL os dados trafegaram por meio de uma rede do tipo *Ethernet 10/100* até a máquina descrita acima antes de serem processados e gravados na nova base de dados. A nova base e a ferramenta OLAP para visualização dos dados foram instaladas no mesmo computador onde o processo de ETL é executado.

As características encontradas no ambiente de implementação do projeto são as seguintes:

- Empresa de pequeno porte;
- Servidor de dados com sistema operacional Linux;
- 4,5 *gigabytes* de dados na base OLTP;
- Banco de dados operacional *Open Source* (*Firebird*);
- Dados históricos na base transacional mantidos a certo período de tempo (desde a implantação dos sistemas informatizados, no ano de 2005);
- Base de dados centralizada para todos os Sistemas de processamento de transações;
- O projeto tem limitação de tempo de oito meses entre estudo das tecnologias, desenvolvimento do projeto, implantação e análise dos resultados, sendo que a carga horária despendida não será integral durante este período.

Com base no levantamento conclui-se:

- Deverá ser analisado o custo de aquisição das ferramentas necessárias para desenvolvimento do projeto;
- As ferramentas utilizadas no processo deverão ser compatíveis com o sistema operacional *Linux*;
- Deverá ser usada uma abordagem que permita a criação de um *Data Warehouse* e obtenção dos resultados no tempo previsto.

Devido ao fato de que uma abordagem global da empresa tomaria muito tempo, e que a arquitetura independente pode se tornar um problema à medida que o número de *Data Marts* cresça, a arquitetura mais apropriada para o presente trabalho é a arquitetura integrada. Na arquitetura escolhida é possível criar *Data Marts* de forma incremental, fazendo com que futuramente os dados de todos os *Data Marts* da empresa apresentem uma visão global dos negócios.

No que diz respeito à abordagem de implementação, concluiu-se que a implementação *bottom-up* é a mais apropriada. Com a possibilidade da criação incremental de *Data Marts* os riscos do projeto são reduzidos e os retornos poderão ser observados em um menor espaço de tempo se comparado com as outras abordagens. Outro fato relevante é que os sistemas utilizados pela empresa não abrangem todos os setores da mesma. Isto traria transtorno na utilização de uma abordagem do tipo *top-down* pois o *Data Warehouse* seria projetado excluindo informações provenientes de outros setores da empresa ainda não informatizados.

De acordo com o usuário final o assunto vendas, que é o principal negócio da empresa, representa a maior necessidade de informações para apoio a tomada de decisão. Portanto, será criado um *Data Mart* com foco no departamento de vendas.

Há uma grande variedade de ferramentas disponíveis para os processos envolvidos na criação e implementação de um *Data Mart* no mercado, porém a maioria delas é proprietária e/ou o seu uso está atrelado a um SGBD específico. Exemplos deste tipo de ferramenta são *Cognos*, *Business Object*, e também as ferramentas fornecidas pelos fabricantes de bancos de dados proprietários, como *Oracle* e *Microsoft* SQL *Server*. Conforme os requisitos descritos anteriormente a melhor solução será o uso de softwares livres, sem custo de aquisição e com integração com bancos de dados não-proprietários.

Os processos de um *Data Warehouse*/*Data Mart* consistem basicamente na obtenção dos dados dos sistemas transacionais, na organização, transformação e integração desses dados em uma outra base de dados e no acesso destes dados para consultas de forma simples, rápida e flexível. Contudo, algumas tarefas são necessárias além destas citadas, como por exemplo, a análise das necessidades do negocio, dos sistemas transacionais e das suas bases de dados, a modelagem multidimensional do *Data Mart*, e a criação dos cubos de dados. Para criação do *Data Mart* foram realizados os seguintes passos, representados pela figura 5 .

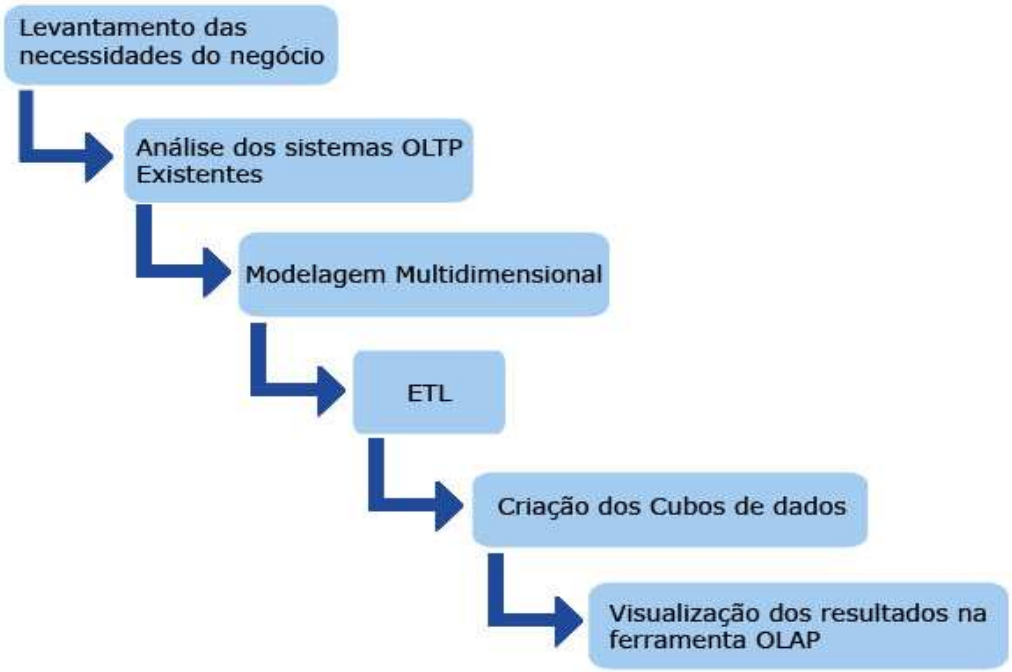



**Figura 5. Estrutura da criação do DM.**

A seguir cada uma das etapas apresentadas na figura 5 será detalhada.

### 4.1.1. Levantamento das necessidades do negócio

Devido a necessidade de se investigar as informações relevantes para a criação do *Data Mart* foi feito um levantamento inicial das necessidades do negócio e a análise dos sistemas transacionais da empresa alvo da aplicação.

Como o objetivo é prover informações para suporte a tomada de decisões em uma empresa de pequeno porte foi definido junto ao usuário final quais seriam as informações relevantes para o novo sistema, e a respeito de quem se tratavam essa informações. A necessidade do cliente foi confrontada com os dados que a empresa possuía em sua base de dados transacional para verificar a viabilidade da criação de um *Data Mart* que atenderia as suas expectativas iniciais.

As informações aqui definidas como necessárias serão apresentadas na seção 4.1.3 deste artigo sob a forma de atributo das tabelas do modelo multidimensional.

### 4.1.2. Análise dos sistemas OLTP existentes.

Com base nas necessidades do cliente foi necessária a realização de uma análise dos dados disponíveis no banco de dados dos sistemas OLTP a fim de concluir se os dados necessários para a criação do *Data Mart* existiam. Foi feita uma análise nas tabelas dos sistemas com o objetivo de encontrar aquelas onde estavam os dados necessários para a obtenção das informações que posteriormente serão usadas pelo DM. Como a base de dados é usada como repositório central para diversas aplicações, com diferentes propósitos, apenas as tabelas com as informações relacionadas aos temas levantados pelo usuário foram analisadas. A figura 6 mostra o MER resumido com as tabelas encontradas.

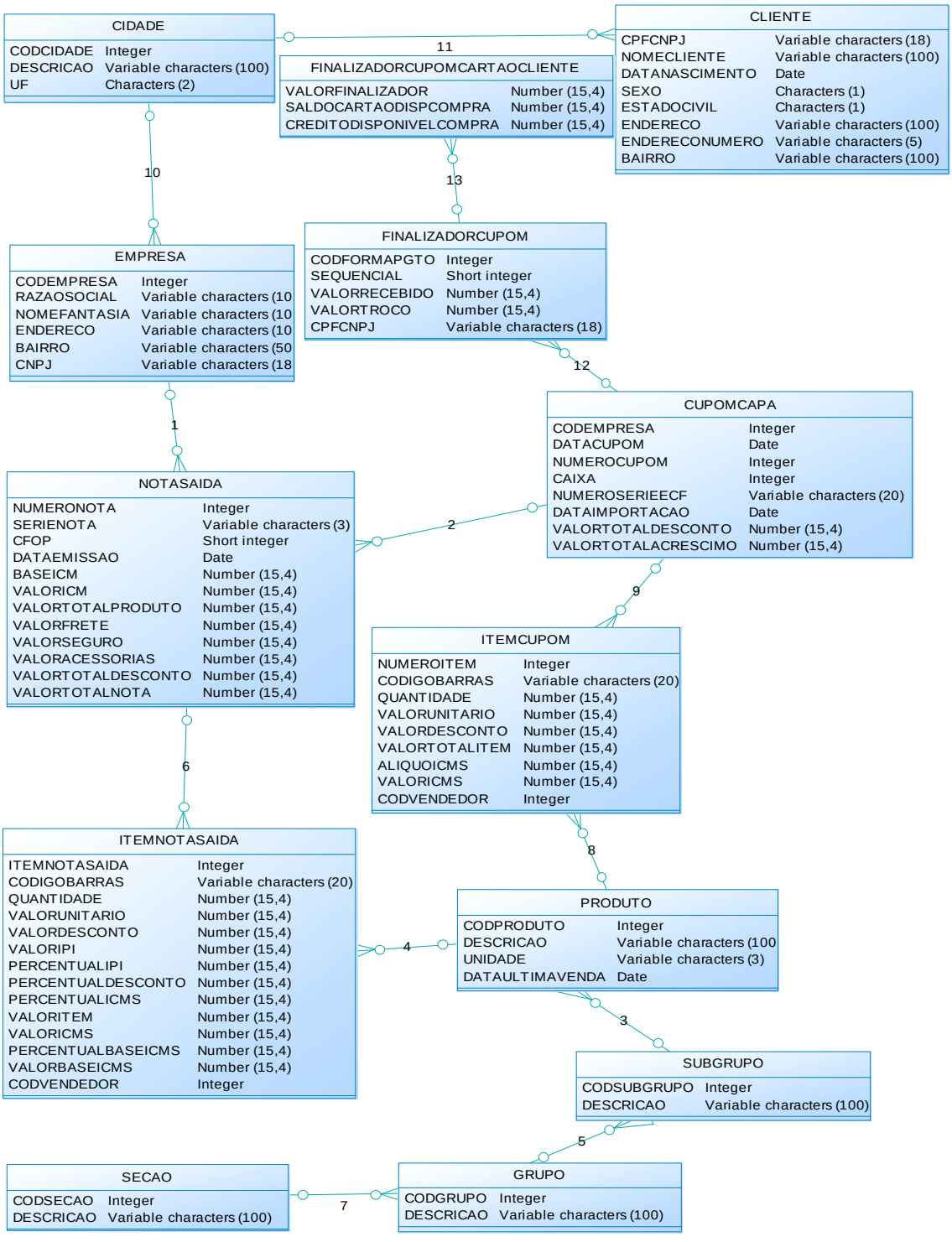



**Figura 6. MER da base OLTP com as tabelas utilizadas.**

### 4.1.3. Modelagem multidimensional

A abordagem utilizada para implementação do projeto tem como princípio a criação de *Data Marts*, incrementalmente, de forma que estes venham a compor um *Data Warehouse* no futuro. Conforme Inmon (2005), o modelo multidimensional é utilizado, exclusivamente, em Data Marts onde é necessária uma estrutura que otimize o acesso a um grande volume de dados. A escolha do modelo multidimensional se deu em virtude

do desempenho do modelo estrela ser superior em 30% ao modelo *SnowFlake* (MACHADO,2000).

O modelo criado para o *Data Mart*, de acordo com as necessidades do cliente e os dados existentes nos sistemas transacionais, consiste em uma tabela de fatos denominada “FATO_VENDAS”, com as medidas de quantidade e valor vendido, e as dimensões “DIM_CLIENTE”, “DIM_EMPRESA”, “DIM_DATA” e “DIM_PRODUTO”.

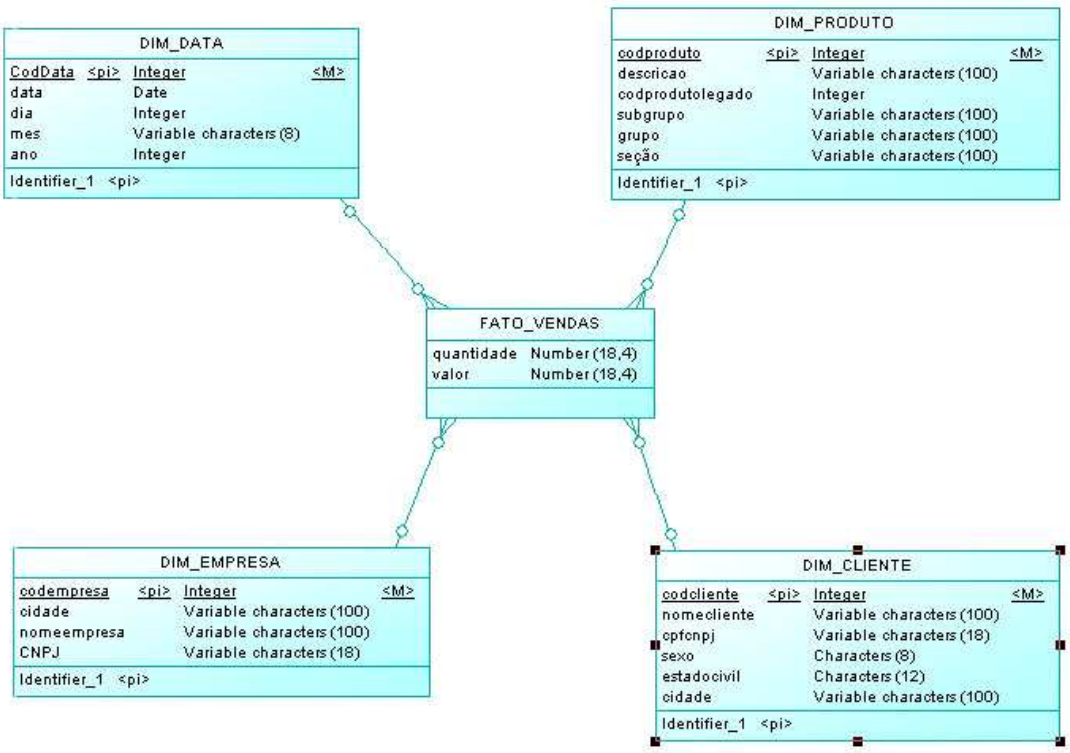



**Figura 7. Modelo de dados multidimensional.**

Para cada uma das tabelas de dimensões deverá ser criada uma chave primária independente da chave da tabela no sistema transacional, geralmente obtida através de um número seqüencial gerado. O uso deste tipo de chave “substituta” previne que ocorram problemas quando ocorrerem mudanças nos dados das dimensões. No caso de alteração em algum dos dados da dimensão é inserido um novo registro, com uma nova chave primária, e com os dados alterados. Para citar como exemplo: a mudança de cidade de um cliente deve resultar em um novo registro na tabela “cliente”, com a nova cidade, mantendo assim os dados históricos do cliente para cada uma das cidades em que ele residiu. A simples atualização do atributo “cidade” na dimensão “DIM_CLIENTE” causaria problemas ao analisar as vendas da cidade onde o cliente residia, pois elas seriam todas movidas para a nova cidade.

### 4.1.4. ETL

Para o processo de ETL do *Data Mart* foi utilizada a ferramenta *Pentaho Data Integration* – PDI – (também conhecida como Kettle). A ferramenta foi escolhida por ser parte de um pacote de ferramentas de BI distribuído pela Pentaho, que abrange

desde o processo de ETL até a apresentação dos dados. A Ferramenta é de código aberto e pode ser executada em diversos sistemas operacionais, tais como *Windows* e *Linux*.

A seguir será mostrado e detalhado um fluxo de transformações da ferramenta PDI para a carga da tabela de fatos (FATO_VENDAS) do DM.

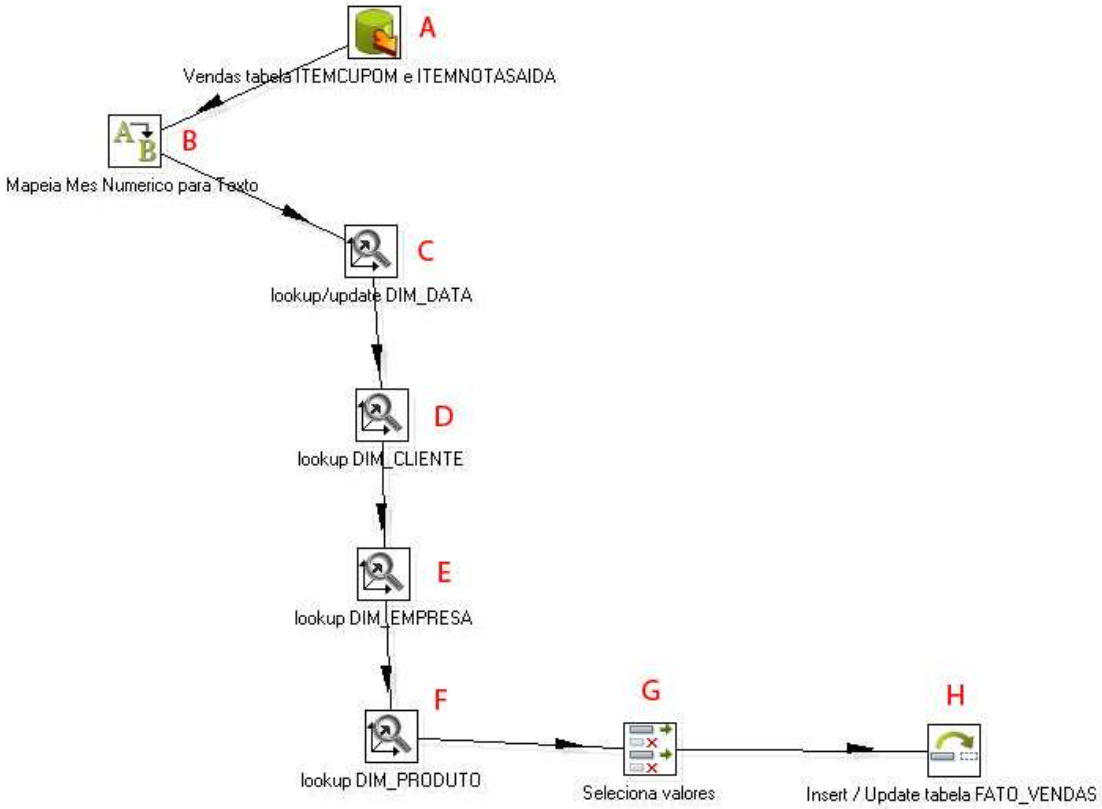



**Figura 8. Fluxo de transformações para carga da tabela fato.**

No fluxo apresentado através da figura 8 existem componentes que adicionam valores, outros transformam, e outros que atualizam os dados na base de dados. A seguir o detalhamento de cada passo da transformação.

Na figura 8, indicada pela letra “A” é exemplificado a carga dos dados relativos as vendas do sistema transacional, onde o componente da ferramenta executa um comando SQL pré determinado. De acordo com o MER do sistema de processamento de transações as tabelas que guardam estes dados são “ITEMCUPOM” e “ITEMNOTASAIDA”. Os valores são carregados do banco de dados do sistema transacional e inseridos no fluxo.

Na figura 8, o componente indicado pela letra “B” faz a conversão do mês da data da venda de um número inteiro de 1 a 12 para um texto equivalente ao nome do mês com 3 caracteres. Este processo é feito pois na visualização dos dados é necessário que os dados se apresentem da forma mais intuitiva possível.

Na figura 8, o componente indicado pela letra “C” procura se a data da venda oriunda do fluxo já existe no banco de dados. Se não existe então é criado um novo registro e é adicionado ao fluxo o valor da chave primária gerada na inserção. Se a data

em análise já existir, apenas é retornado o valor da chave da dimensão “DIM_DATA” equivalente àquela data específica e é adicionado ao fluxo.

Na figura 8, o componente indicado pela letra “D” procura na dimensão dim_cliente do DM o valor da chave substituta (a chave usada pelo DM) para um determinado CPF/CNPJ do fluxo de dados.

Na figura 8, o componente indicado pela letra “E” procura na dimensão dim_empresa do DM o valor da chave substituta equivalente a determinado CNPJ da empresa do fluxo de dados.

Na figura 8, o componente indicado pela letra “F” procura na dimensão dim_produto do DM o valor da chave substituta que equivale ao produto de código igual ao “codigoprodutolegado” advindo do fluxo de dados.

Na figura 8, o componente indicado pela letra “G” atua como um filtro nos valores do fluxo. Podem ser removidos campos que estão no fluxo que não serão usados nos passos seguintes.

Na figura 8, o componente indicado pela letra “H” representa a inserção dos dados do fluxo na tabela de fatos do DM. Este mesmo passo insere as chaves substitutas encontradas nos passos anteriores juntamente com as medidas. No caso dos dados já estarem no DM eles serão ignorados.

Como as tabelas das dimensões sofrem poucas alterações, foram feitos fluxos de transformação para cada tabela de dimensão, onde são carregados os dados dessas tabelas. Na execução da transformação exemplificada na figura 8 supõe-se que os dados das dimensões já foram carregados no DM. Já a dimensão “DIM_DATA” será atualizada juntamente na carga da tabela de fatos visto que ela não necessita de carga prévia por não ser uma tabela do sistema transacional.

Para a execução das transformações pode ser criado um “*job*”, que executa seqüencialmente as transformações podendo ser agendado um horário para execução automática. No fluxo demonstrado na figura 9 é feita a carga das dimensões, uma a uma, com exceção da dimensão “DIM_DATA”, e no penúltimo passo é feita a carga da tabela “FATO_VENDAS”. No final do processo é enviado um e-mail contendo o *log* das operações realizadas.

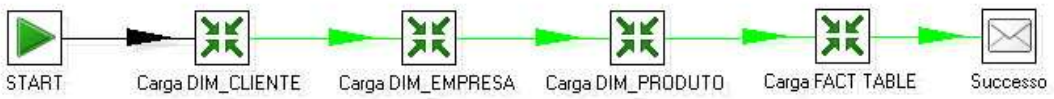



**Figura 9. Fluxo de transformações executadas por um “job”.**

### 4.1.5. Criação dos Cubos de dados.

Para a criação do cubo de dados foi usada a ferramenta *Mondrian Schema Workbench* (MSW), a qual permite criar e testar esquemas de cubos OLAP através de uma interface gráfica. O *schema* gerado pela ferramenta é utilizado para definição do cubo de dados dentro do servidor OLAP *Mondrian*, que é um produto da suíte BI da pentaho. O

servidor OLAP executa consultas MDX [1] em uma base de dados relacional e apresenta os resultados em um formato multidimensional. A estrutura de uma consulta MDX se parece com o código apresentado no quadro 2.

```
SELECT {[Measures].[Unit Sales], [Measures].[Store Sales]} ON COLUMNS,
  {[Product].members} ON ROWS
FROM [Sales]
WHERE [Time].[1997].[Q2]
```

**Quadro 2. Exemplo de consulta MDX.**

A linguagem de consulta MDX é semelhante a linguagem SQL usada nos bancos de dados relacionais. MDX é um padrão, porém este artigo cobre apenas o dialeto do *Mondrian* que, apesar de ser específico para este servidor OLAP, foi feito com base na linguagem MDX padrão.

A ferramenta MSW cria a definição do cubo OLAP. Para cada novo *schema* criado é possível definir cubos, e dentro de cada cubo sua tabela de fatos, medidas e dimensões. Na ferramenta também é especificada a hierarquia dos atributos de cada dimensão, que serão utilizadas para operações de *Drill-down* e *Drill-up* posteriormente.

A figura 10 representa uma tela de exemplo de *schema* de cubo criado na ferramenta *Mondrian Schema Workbench*.

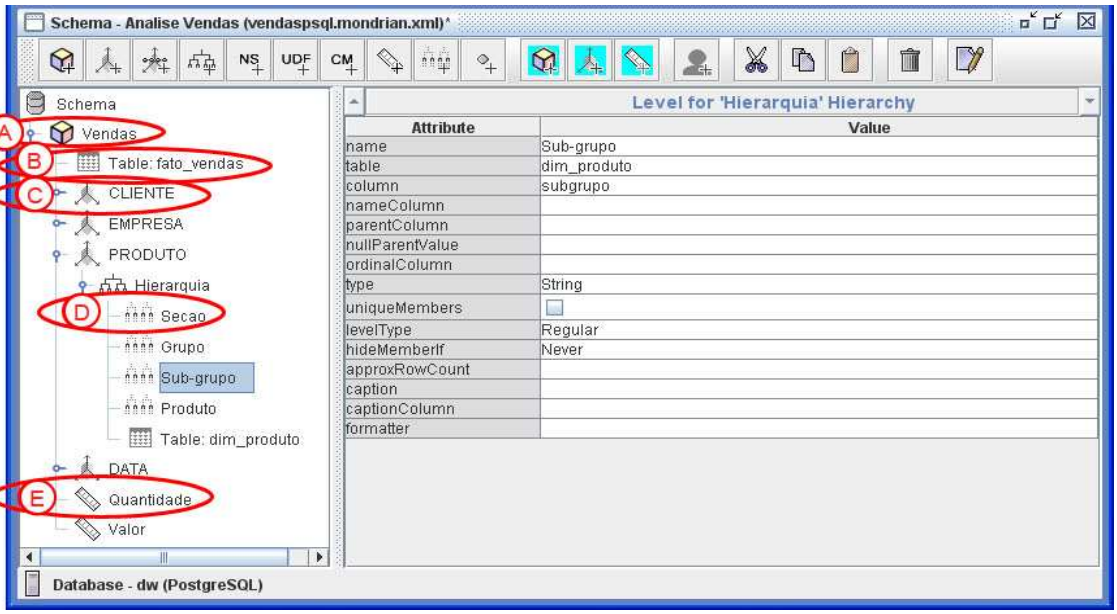



**Figura 10. Exemplo de tela da ferramenta MSW.**

É possível ver através da figura 10 o detalhamento da estrutura do cubo, na ferramenta.

Na figura 10, no item indicado pela letra “A” está representada a estrutura criada para o cubo. Todos os itens mostrados, na seqüência, pertencem ao hipercubo

[1] MDX é uma linguagem de consulta que significa *multi-dimensional expressions* – expressões multidimensionais – e é usada como linguagem para consultas multidimensionais no servidor OLAP Mondrian.

“Vendas”. Para cada cubo são definidas a tabela de fato (item “B” na figura), as suas dimensões (item “C”), membros das dimensões hierarquizados (item “D”) e medidas (item “E”).

O item indicado pela letra “C” na figura 10 representa uma dimensão do modelo multidimensional. Para cada dimensão são definidos os membros, hierarquia e a tabela da dimensão no banco de dados.

O item indicado pela letra “D” na figura 10 representa um membro de uma hierarquia. A ordem dos níveis da hierarquia são definidos conforme a ordem em que estão dispostos. De acordo com a figura 10 a dimensão “Produto” terá os membros hierarquizados da seguinte forma, do nível menos detalhado para o mais detalhado: Seção, grupo, sub-grupo e produto.

O item representado pelo item “E” na figura 10 representa uma medida da tabela fato. No exemplo da figura existem duas medidas : Quantidade e valor.

Apesar da ferramenta apresentar uma interface gráfica para manipular as estruturas do cubo, o arquivo que contêm o *schema* do modelo gerado é salvo no formato XML[2], portanto pode ser editado com qualquer editor de textos. A sintaxe de definição é apresentada na quadro 3.

```
<Schema name="Analise Vendas">
  <Cube name="Vendas" cache="true" enabled="true">
    <Table name="fato_vendas" schema="public"></Table>
    <Dimension type="StandardDimension" foreignKey="codproduto" name="PRODUTO">
      <Hierarchy name="Hierarquia" hasAll="true" allMemberName="Todos os Produtos"
                 primaryKey="codproduto">
        <Table name="dim_produto" schema="public"></Table>
        <Level name="Secao" table="dim_produto" column="secao" type="String"
               uniqueMembers="false" levelType="Regular" hideMemberIf="Never"></Level>
        <Level name="Grupo" table="dim_produto" column="grupo" type="String"
               uniqueMembers="false" levelType="Regular" hideMemberIf="Never"></Level>
        <Level name="Sub-grupo" table="dim_produto" column="subgrupo" type="String"
               uniqueMembers="false" levelType="Regular" hideMemberIf="Never"></Level>
        <Level name="Produto" table="dim_produto" column="descricao" type="String"
               uniqueMembers="false" levelType="Regular" hideMemberIf="Never"></Level>
      </Hierarchy>
    </Dimension>
    ...
    ...
    <Measure name="Quantidade" column="quantidade" datatype="Numeric"
             formatString="#,###" aggregator="sum" visible="true"></Measure>
    <Measure name="Valor" column="valor" datatype="Numeric"
             formatString="#,###" aggregator="sum" visible="true"></Measure>
  </Cube>
</Schema>
```

**Quadro 3. Exemplo de um schema de cubo de dados.**

### 4.1.6. Visualização dos resultados na ferramenta OLAP

Como a estrutura do cubo foi criada a partir da ferramenta *Mondrian Schema Workbench*, será usado para visualização dos dados o produto da suíte BI da pentaho chamado BI *Serve*”, o qual utiliza o servidor OLAP *Mondrian*. A ferramenta é desenvolvida como uma aplicação web *Java*, logo é necessário a utilização de um servidor web compatível com essa tecnologia, tais como *Tomcat* e *JBoss*. Para desenvolvimento do trabalho foi utilizada a versão 3.0 do BI *Server* juntamente com o *Apache Tomcat 5.5.26*.

[2] eXtensible Markup Language

Antes de visualizar as informações foi necessário enviar o arquivo contendo as informações sobre o cubo de dados para o BI *Server* e configurar a ferramenta OLAP para que ela pudesse se conectar com o banco de dados criado anteriormente para armazenar os dados do *Data Mart*. A publicação do cubo no BI *Server* pode ser feita pela própria ferramenta *Mondrian Schema Workbench*, pela opção "*Publish*", no menu "*File*".

A visualização dos dados dentro da ferramenta é feita pelo menu "Analysis *View*", que apresenta, sob a forma de uma tabela, os dados do cubo criado, podendo ser feitas diversas operações tais como *drill-up*, *drill-down*, *slice*, *dice*. Há ainda a possibilidade de gerar gráficos dos dados apresentados na tabela e customizar o formato de apresentação do gráfico, tamanho, cores. A geração dos gráficos é feita em tempo real de acordo com os dados apresentados na tabela acima citada. Na ocorrência de uma mudança nos dados apresentados, como por exemplo, numa operação para detalhamento das vendas por data, o gráfico gerado automaticamente muda para comportar a nova apresentação dos dados.

Ainda é possível exportar a tabela com a análise dos dados para o formato "*.xls*", padrão do editor de planilhas *Excel* e também utilizado por outros aplicativos processadores de planilhas, tornando assim mais flexível a manipulação das informações obtidas.

A partir do modelo multidimensional e também da definição do cubo criado, foi possível visualizar as informações através da ferramenta BI *Server*, da forma apresentada na figura 11. É possível visualizar o cubo de dados com as informações a respeito do volume de vendas do ano de 2009, para clientes da cidade de Guaporé, de dois subgrupos de produtos.


**Figura 11. Exemplo de tela do BI Server mostrando as informações do cubo.**

Sobre esta mesma perspectiva realizou-se uma operação de *Drill down* sobre o membro "ano" e um *Slice*, mostrando apenas a fatia de dados do sub-grupo "AVIAMENTOS". Os resultados destas mudanças de perspectiva podem ser observados juntamente com um gráfico comparativo do volume de vendas nos meses do ano na figura 12.

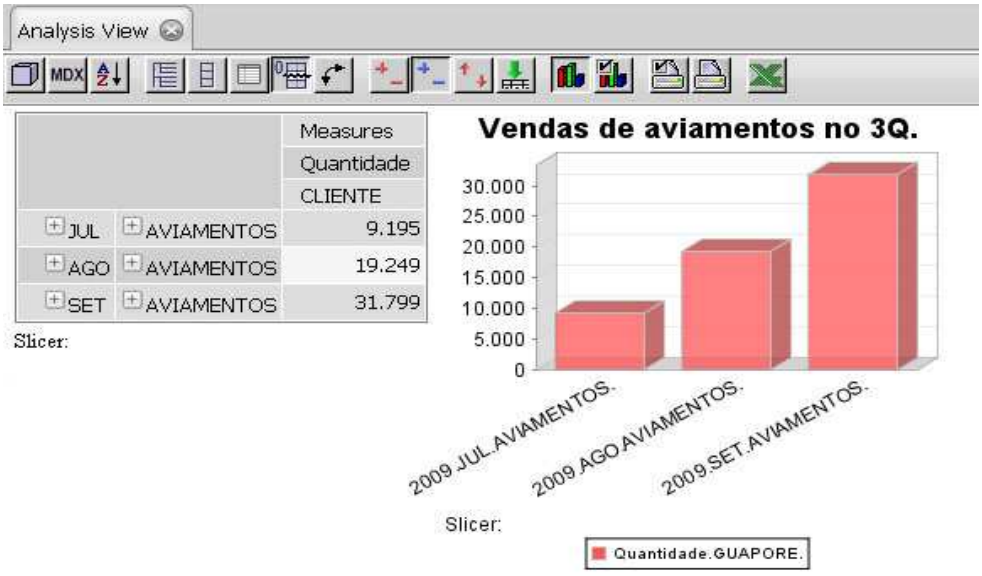



**Figura 12. Exemplo de tela do BI Server com a geração de um gráfico.**

Estes dois exemplos apenas ilustram o que pode ser feito com a ferramenta. As mesmas operações de navegação através do nível de detalhe e mudança de perspectiva podem ser feitas sobre qualquer outra análise dos dados.

## 5. Conclusões

Neste trabalho realizou-se uma pesquisa acerca de métodos e ferramentas para a criação de um projeto de *Data Warehouse*. A arquitetura de *Data Mart* integrado, e a abordagem de implementação *bottom-up* foram escolhidas devido ao tempo limitado para desenvolvimento e implantação do projeto, e também por não ser possível obter dados de toda a empresa em virtude da não informatização de alguns setores. Estas técnicas visam a criação de um *Data Warehouse*, de forma incremental, a partir de *Data Marts* interconectados. Embora esta abordagem possa apresentar algumas desvantagens, já que pode gerar um problema de integração no futuro, ela foi a mais adequada aos requisitos levantados no ambiente de implementação.

Como exemplo de aplicação, realizou-se um mapeamento das necessidades de uma empresa de pequeno porte da região, com o intuito de obter informações relevantes para o suporte a decisão nos níveis mais altos de sua organização. Devido a realidade encontrada neste ambiente utilizou-se de ferramentas software livre durante o processo de criação do cubo de dados, ETL e visualização dos dados carregados no *Data Mart*. Estas ferramentas além de possuírem um nível de integração entre elas, facilitando o uso, não possuem custo de aquisição.

Utilizou-se como fonte de dados a área de vendas da empresa, considerando que esta era a área onde, segundo o usuário final, mais se necessitava de informações voltadas ao processo de tomada de decisão. O modelo de dados multidimensional projetado para o presente artigo possui apenas as dimensões básicas descritas na literatura, no entanto as informações obtidas através das consultas realizadas no servidor OLAP foram satisfatórias, pois atendem aos requisitos inicias dos futuros usuários do SAD. A partir destas informações será possível tomar decisões levando em conta os fatos ocorridos na empresa ao longo dos anos.

Com base no presente estudo é possível verificar que novos requisitos surgirão ao longo do tempo, e novos *Data Marts* serão criados para outros propósitos, compondo um repositório central dos dados de todos os setores da empresa.

Durante o período de 8 meses onde foi estudado, desenvolvido e aplicado o projeto do *Data Mart*, foram encontradas algumas dificuldades. As principais foram:

- Nem todos os setores da empresa dispunham de sistemas informatizados, o que dificultava a visão de um *Data Warehouse* com abrangência global;
- Pouco tempo para desenvolvimento, implantação e análise dos resultados do projeto, considerando que estas etapas levaram 5 meses para serem concluídas.

Estas dificuldades impossibilitaram a criação de um projeto de *Data Warehouse* mais detalhado no início. Entretanto é necessário destacar que a criação de um *Data Warehouse* a partir de *Data Marts* interconectados não está descartada. A utilização deste enfoque esta mais de acordo com o ambiente de implementação, pois atualmente não é possível obter dados de todos os setores da empresa.

Como trabalho futuro, além da criação de outros *Data Marts* voltados aos demais departamentos da empresa, um estudo detalhado de ferramentas open-source disponíveis no mercado seria válido. As ferramentas citadas no presente artigo ajudaram no decorrer do projeto, porém elas foram analisadas somente em relação à integração com outras ferramentas utilizadas no processo desenvolvimento. Um estudo do impacto do uso do sistema na empresa também poderá ser feito no futuro, a fim de comprovar os benefícios do utilização de um SAD.

## Bibliografia


Inmon, W. H. (2005). *Building the Data Warehouse.* Fourth Edition. Wiley Publishing, Inc, USA

Inmon, W. H. (1997). *Como Construir o Data Warehouse.* Campus, Rio de Janeiro

Laudon, K. C. e Laudon, J. P. (2007). *Sistemas de informação gerenciais*. Pearson Prentice Hall, São Paulo

Machado, F. N. R. (2000). *Projeto de Data Warehouse: Uma Visão Multidimensional*. Érica, São Paulo.

PENTAHO COMMUNITY. Pentaho Wiki. *Latest Pentaho Data Integration (aka Kettle) Documentation*. Disponível em: < http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+(aka+Kettle)+Documentation>. Acesso em ago. 2009.

PENTAHO COMMUNITY. Pentaho Wiki. *BI Server 2.x-3.x Community Documentation*. Disponível em: < http://wiki.pentaho.com/display/ServerDoc2x/BI+Server+2.x-3.x+Community+Documentation>. Acesso em ago. 2009.

PENTAHO COMERCIAL OPEN SOURCE BUSINESS INTELIGENCE. *Mondrian Overview*. Disponível em: < http://mondrian.pentaho.org/documentation/doc.php> . Acesso em set. 2009.