**UNIVERSIDADE FEDERAL DE SANTA CATARINA**
**PROGRAMA DE PÓS-GRADUAÇÃO**
**EM CIÊNCIA DA COMPUTAÇÃO**

**Sérgio Luis Dill**

# UMA METODOLOGIA PARA DESENVOLVIMENTO DE DATA WAREHOUSE E ESTUDO DE CASO

Dissertação submetida à Universidade Federal de Santa Catarina como requisito final para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Murilo Silva de Camargo

Florianópolis, Outubro de 2002.

# UMA METODOLOGIA PARA DESENVOLVIMENTO DE DATA WAREHOUSE E ESTUDO DE CASO

**Sérgio Luis Dill**

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____
Prof. Fernando Álvaro Ostuni Gauthier, Dr.
Coordenador do Curso

Banca Examinadora

_____
Prof. Murilo Silva de Camargo, Dr. - Orientador

_____
Prof. Rosvelter João Coelho da Costa, Dr. – INE, UFSC

_____
Prof. Vitório Bruno Mazzola, Dr. – INE, UFSC

_____
Prof. Roberto Willrich, Dr. – INE, UFSC

A Deus pela vida...

Aos meus pais pelos exemplos,

À minha esposa Elegiane pela companhia e apoio,

Aos meus filhos Guilherme e Poliana por tudo que representam para mim.

# AGRADECIMENTOS

A Deus, fonte de toda vida.

A UNIJUI – Universidade Regional de Ijuí, pela oportunidade concedida para a realização deste curso.

A SETREM – Sociedade Educacional Três de Maio pelo trabalho e esforço na implantação do curso.

Aos professores da UFSC pelos ensinamentos transmitidos.

Ao professor Murilo Silva de Camargo, pela orientação e ensinamentos transmitidos.

Aos colegas da Coordenadoria de Informática da UNIJUÍ pela amizade, estímulo e contribuições.

Finalmente a todos aqueles que de uma forma ou de outra deram a sua contribuição neste trabalho.

# RESUMO

O ambiente de *data warehouse* (DW) surgiu como uma evolução dos ambientes de suporte a decisão, integrando fontes de dados dos sistemas transacionais. Sua crescente popularidade reflete a necessidade das empresas em obter informações analíticas derivadas dos seus sistemas transacionais. O ambiente de *data warehouse* tem características diferentes do ambiente tradicional e é construído com o objetivo de suprir as necessidades de processamento analítico das organizações. Os projetos de *data warehouse* têm mais chances de sucesso quando desenvolvidos através de uma metodologia consistente que identifique e guie o projetista durante as várias fases do projeto.

O objetivo principal deste trabalho é elaborar uma metodologia para desenvolvimento de *data warehouse*. O aspecto principal a ser contemplado na elaboração desta metodologia e a sua aplicabilidade prática. Assim, é de fundamental importância que a metodologia seja suportada por uma ferramenta de desenvolvimento. Outro aspecto importante na elaboração da metodologia é a clara identificação das várias fases do processo de construção do data warehouse com a finalidade de guiar o projetista ao longo do projeto.

Posteriormente, essa metodologia proposta, foi utilizada para desenvolver o estudo de caso desta dissertação. O estudo de caso tem a finalidade de verificar e avaliar a aplicabilidade da metodologia proposta. Para o estudo de caso foi utilizado o sistema de Concurso Vestibular da UNIJUÍ sediada na cidade de Ijuí (RS).

# ABSTRACT

Data Warehousing appeared as the natural evolution of decision support systems, integrating information sources from transactional corporative systems. The increasing popularity of *data warehouse* systems reflects the enterprise needs of analytical information derived from their transactional systems. Such systems are mostly developed over the well known three level architecture (conceptual modeling, logical project and physical project) widely used which benefits performance, security and data integrity. *Data warehouse* environment has different characteristics than transactional systems and is built having in mind the requirements of analytical processing needs of the organization and like transactional systems, the *data warehouse* systems have more choices to be successfully developed thorough the use of a consistent methodology that identify the different phases of the project.

The main objective of this work is to develop a design methodology for data warehouse development. The main issue to be addressed in the methodology is its practical applicability. Thus, it is mainly important that the design methodology to be supported by a data warehouse design tool. Another issue in the methodology is a clear distinction between the several fases os the data warehouse design process in order to guide the designer through the overall process.

In last, the methodology has been used to developed the case study of this job. The case study was be developed to verify and evaluate the practical applicability of the methodology. In order to developed the case study, we used a subset of data extracted from the operational databases of UNIJUÍ located at IJUÍ (RS).

# SUMÁRIO

# LISTA DE FIGURAS

# LISTA DE TABELAS

# 1. INTRODUÇÃO

## 1.1. Posicionamento

Nas duas últimas décadas temos acompanhado uma evolução tecnológica bastante acelerada. As empresas fabricantes de *hardware* têm continuamente lançado produtos novos cada vez mais poderosos aumentando a capacidade de processamento das máquinas e seus periféricos. Paralelamente, a indústria de *software* está explorando ao máximo toda a evolução do *hardware* e através de sucessivas atualizações através de novas versões de seus produtos incrementam o espectro de soluções disponíveis para as empresas. Um exemplo disso é a heterogeneidade de Sistemas Gerenciadores de Banco de Dados (SGBDs) disponíveis no mercado, produtos de excelente qualidade oferecendo uma gama variada de recursos, facilitando o trabalho dos profissionais envolvidos na administração do crescente volume de dados presente nas empresas.

Por outro lado, a competitividade cada vez mais acirrada neste mundo em processo de globalização, faz com que as empresas cada vez mais invistam na capacidade de geração de informação a partir dos dados presentes nos seus sistemas. A geração de informação tornou-se um elemento fundamental para conseguir um diferencial tecnológico e econômico num mercado cada vez mais competitivo em todas as esferas (bens, serviços, pesquisa, educação, desenvolvimento).

Conforme INMON (1997), o processo de geração de informações para auxiliar na tomada de decisões não é uma atividade nova. No início da década de 1960, o mundo da

computação consistia na criação de aplicações individuais que eram executadas sobre arquivos mestres. A proliferação de arquivos mestres e a massiva redundância não controlada de dados conseqüentes da arquitetura de desenvolvimento espontâneo geraram problemas tais como:

- Falta de credibilidade dos dados;
- Baixa produtividade;
- Impossibilidade de transformar dados em informações.

Com advento dos SGBDs e a conseqüente integração do dados em uma fonte única para todo o processamento, surgem os primeiros Sistemas de Apoio a Decisão (SAD). Com a necessidade crescente de sistemas de suporte à decisão caracterizado pelo novo ambiente de negócios surgiu a necessidade de separar o ambiente de processamento operacional do ambiente analítico. Desta evolução natural surge em 1992 o conceito de *Data Warehouse* como requisito aos emergentes sistemas informacionais. Posteriormente surgiu o conceito de *Data Warehousing* abrangendo o conjunto de tecnologias empregadas nestes ambientes (INMON, 1997).

## 1.2. Justificativa para este trabalho

Atualmente, em muitas empresas, o processo manual de produção de informações gerenciais é demorado, dispendioso e cansativo, pois reúne uma grande quantidade de dados que precisam ser coletados de diversas fontes e convertidos em num formato apropriado que possibilite a sua análise. A criação de um ambiente de data warehouse surgiu como uma alternativa viável cujo princípio está na criação de um banco de dados especializado capaz de manipular grande volume de informações com bom desempenho, melhorando a gerência, o controle e o acesso aos dados. A função do *data warehouse* é tornar as informações corporativas, obtidas a partir de bancos de dados operacionais e de fontes de dados externas à organização, acessíveis para entendimento e uso das áreas estratégicas de uma organização.

A criação de um ambiente de *data warehouse* automatiza o processo de geração de informações gerenciais com as seguintes vantagens:

- Atualização constante;
- Agilidade e facilidade de acesso à informação através de uma ferramento OLAP;
- Menor custo e menor índice de erros.

No entanto, um projeto de *data warehouse* é uma tarefa complexa envolvendo um conjunto de conceitos e tecnologias. O sucesso de um projeto de *data warehouse* está estreitamente relacionado com o entendimento e domínio destes conceitos e tecnologias. De acordo com KELLY (1997), a causa principal que resulta em falha e insucesso de um projeto de data warehouse está relacionada à ausência de uma metodologia abrangente capaz de fornecer uma visão geral do processo envolvendo estes conceitos e tecnologias.

### 1.2.1. Objetivo geral

O objetivo principal deste trabalho é elaborar uma metodologia para desenvolvimento de *data warehouse*. Atualmente, a bibliografia carece de uma metodologia consistente e de provada aplicabilidade prática. No capítulo 5, apresenta-se três metodologias identficadas na literatura. Ao final do capítulo 5, avalia-se estas metodologias identificando seus pontos positivos e negativos.

O aspecto principal a ser contemplado na elaboração desta metodologia e a sua aplicabilidade prática. Assim, é de fundamental importância que a metodologia seja suportada por uma ferramenta de desenvolvimento.

Outro aspecto importante na elaboração da metodologia é a clara identificação das várias fases do processo de construção do data warehouse com a finalidade de guiar o projetista ao longo do projeto.

### 1.2.2. Objetivos específicos

Como objetivos específicos deste trabalho podemos destacar:

- Criação de um ambiente de *data warehouse* com a finalidade de oferecer a UNIJUÍ – Universidade Regional do Noroeste do Estado do Rio Grande do Sul, um ambiente de dados integrado e que atenda as necessidades de informações gerenciais da Universidade.
- Desenvolver um estudo de caso para avaliar a metodologia utilizada;
- Adquirir experiência na tecnologia de *data warehousing*;
- Identificar as potencialidades e dificuldades que um sistema de *data warehouse* oferece.
- Separação dos dados do ambiente operacional que processa as transações do dia-a-dia para o ambiente de *data warehouse*.

## 1.3. Organização do trabalho

Este trabalho está dividido em sete capítulos incluindo esta introdução. Inicialmente no item posicionamento do trabalho, aborda-se sucintamente os aspectos que influenciaram na evolução e surgimento dos sistemas de informação. Apresenta-se a crescente influência e valor que informação representa para no ambiente de atuação das empresas.

O Capítulo 2 trabalha os conceitos relacionados ao ambiente de *data warehouse* e os aspectos que influenciaram o seu surgimento. Também discutem-se as formas de implementação do *data warehouse*, as vantagens e desvantagens que cada abordagem oferece. Neste capítulo apresenta-se uma arquitetura de *data warehouse* genérica. A partir da arquitetura genérica, poderão existir várias derivações cada qual com características específicas e propósitos distintos de acordo com a necessidade da organização.

No Capítulo 3 apresenta-se os aspectos da modelagem de dados para *data warehouse*. Maior ênfase é dada para a abordagem da modelagem dimensional como alternativa a já popular modelagem ER. Discutem-se as vantagens da modelagem dimensional no ambiente de *data warehouse*. O projeto de dados no modelo dimensional permite que operações de análise e exploração de dados sejam efetuadas de maneira interativa.

O Capítulo 4 aborda as ferramentas de acesso ao *data warehouse*. Conhecidas como ferramentas OLAP representam, juntamente como o banco de dados, os dois principais componentes do ambiente de *data warehouse*. Através da análise das características e funcionalidades que cada ferramenta oferece pode-se decidir pela melhor alternativa para aquisição da ferramenta desejada. Com o objetivo de tornar o processo de escolha de uma ferramenta OLAP mais simples e compreensiva, enumera-se as diferenças entre o processamento analítico e o processamento transacional.

O Capítulo 5 apresenta três metodologias para projeto e desenvolvimento de *data warehouse*. O capítulo apresenta também uma avaliação das metodologias apresentadas destacando seus pontos positivos e negativos.

A partir da decisão de não utilizar integralmente nenhuma das metodologias apresentadas como base para o estudo de caso, optou-se pela elaboração de uma metodologia que suprisse os pontos críticos das metodologias apresentadas. Desta forma, o Capítulo 6, apresenta uma metodologia que complementa as anteriores, detalhando os aspectos não contemplados pelas metodologias apresentadas anteriormente, com destaque a fase de levantamento de requisitos do *data warehouse* e modelagem conceitual.

O Capítulo 7 apresenta o estudo de caso desenvolvido com base na metodologia descrita no capítulo 6. O estudo de caso usa o sistema de concurso vestibular da UNIJUÍ – Universidade Regional do Noroeste do Estado do RGS como exemplo.

# 2. AMBIENTE DE DATA WAREHOUSE

Este capítulo apresenta uma visão geral do ambiente de *data warehouse* e os fatores que influenciaram o seu surgimento e desenvolvimento. Segue-se com a definição de *data warehouse* conforme a visão de alguns dos principais autores e as caracterísitcas que distinguem um ambiente de *data warehouse* de um ambiente operacional. Apresentamos também as razões que justificam a implementação de um sistema de *data warehouse*. Ao final do capítulo, abordamos diferentes arquiteturas de *data warehouse* e as opções de implementação.

## 2.1. A evolução dos sistemas de apoio à decisão

A história do *data warehouse* está diretamente ligada à evolução de vários aspectos que tiveram impacto decisivo nas empresas. O desenvolvimento de uma nova tecnologia de armazenamento e acesso (DASD) como alternativa à tradicional forma de armazenamento em fita magnética possibilitou o acesso direto aos dados em substituição ao acesso seqüencial imposto pela tecnologia de armazenamento em fita magnética. O surgimento dos sistemas de gerenciamento de banco de dados que se tornaram produtos comerciais populares a partir do início da década de 1980. A criação do modelo relacional como sua simplicidade juntamente com a capacidade de consulta provida pela linguagem SQL.

Com o advento dos computadores pessoais (PC's) e das linguagens de quarta geração (L4Gs) o usuário final percebeu que era possível utilizar os dados para outros

objetivos além do processamento das transações do dia-a-dia efetuados no banco de dados on line. Surgem então os Sistemas de Informações Gerenciais atualmente conhecidos como Sistemas de Apoio a Decisão. Estes sistemas consistiam em processamento com o objetivo de auxiliar no processo de tomada de decisão. Os usuários começaram a construir suas próprias aplicações. Os dados eram extraídos do banco de dados central e manipulados através de planilhas eletrônicas ou aplicativos que eram criados especificamente para este propósito. A principal razão da extração dos dados do ambiente operacional era para evitar o impacto na performance causado pelos programas de análise. Conforme (INMON,1997), o processo indiscriminado de extração de dados trouxe vários problemas:

- Falta de credibilidade dos dados, conseqüente de extrações feitas de fontes de dados diferentes;
- Baixa produtividade causada pele necessidade de analisar *layout's* de vários arquivos para efetuar a extração dos dados desejados;
- Dificuldade de gerar informações a partir dos dados extraídos. Esta dificuldade é resultante da baixa integração entre os dados extraídos de fontes diversas;
- Os usuários finais estavam insatisfeitos com o tempo requerido para desenvolver um aplicativo de relatórios estratégicos e para adequá-lo às novas exigências;

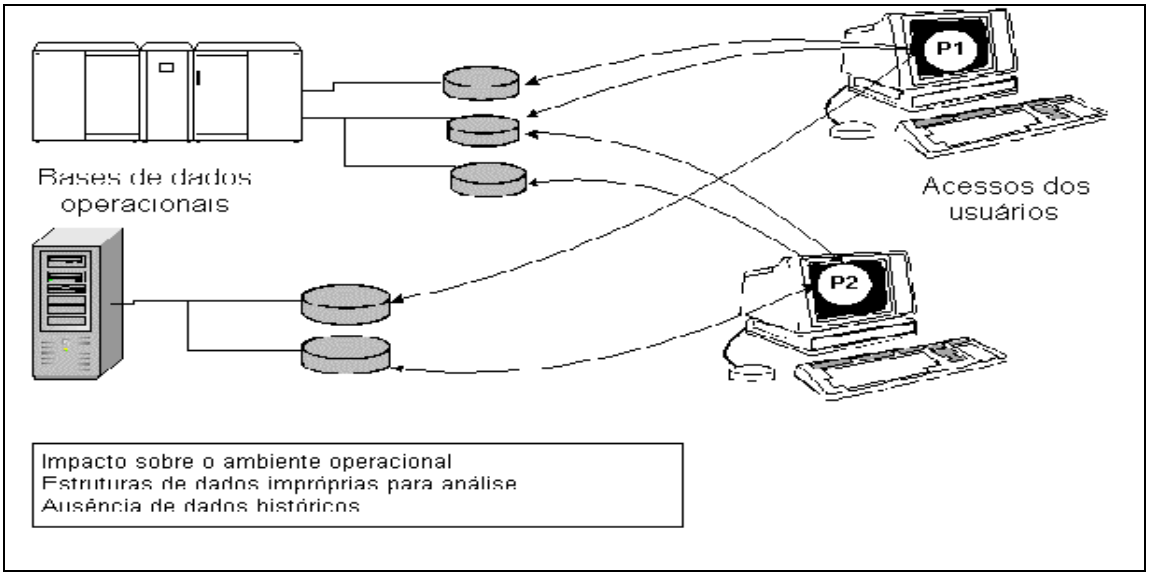



FIGURA 2.1: O ambiente tradicional

A figura 2.1 mostra o ambiente tradicional no qual os usuários executavam os processos de análise de dados diretamente sobre os dados armazenados nos bancos de dados operacionais. As principais dificuldades decorrentes do ambiente tradicional no qual eram desenvolvidos os sistemas de apoio à decisão eram: o impacto que as operações de análise causavam sobre o ambiente operacional que processava as transações do dia a dia da empresa, as estruturas de dados projetadas não eram adequadas ao processo de análise de dados. Outra dificuldade existente neste ambiente era a ausência de dados históricos. Os sistemas transacionais geralmente removem dados antigos com a finalidade de tornar os ambientes mais compactos melhorando o tempo de resposta das aplicações.

Os esforços empreendidos na busca da solução dos problemas aliada ao surgimento de novas ferramentas que possibilitassem soluções mais abrangentes, flexíveis, eficientes e econômicas originaram a nova tecnologia hoje conhecida como "*data warehouse*". O *data warehouse* é a busca da integração de diversos recursos e experimentos ao longo das últimas duas décadas. Esses experimentos permitiram a indústria identificar os principais problemas que necessitavam de solução. O antigo modelo baseado nos sistemas de informações gerenciais, o qual limitava-se à geração de relatórios baseados nas extrações dos dados, não era mais aceito pelas organizações. Esse modelo de desenvolvimento não atendia mais as necessidades das organizações. Fez-se necessário uma mudança de arquitetura, uma mudança de enfoque. Esta mudança de enfoque surgiu da percepção de que há fundamentalmente duas espécies de dados. Os dados primitivos e os dados derivados. Os dados primitivos são dados detalhados manipulados pelas aplicações que automatizavam os processos do dia-a-dia da vida da organização. Os dados derivados são dados resultantes das extrações e eram resumidos ou tabulados para atender às necessidades específicas da gerência. Desta percepção, surge a necessidade de separar o ambiente de dados primitivos do ambiente de dados derivados. O ambiente de dados primitivos deve atender a comunidade funcional. Dados derivados atendem a atividade gerencial. Desta mudança de arquitetura surge o novo ambiente denominado de ambiente projetado de *Data warehouse*.

## 2.2. Definição de data warehouse

Em INMON (1997) encontramos a definição de *warehouse* como sendo um banco de dados baseado em assuntos, integrado, não-volátil e variável em relação ao tempo que é usado principalmente no processo de tomada de decisões.

Segundo KIMBALL (1998), um *data warehouse* é uma copia dos dados especialmente estruturados para facilitar o processo de análise, consulta e geração de relatórios.

Uma definição mais abrangente encontra-se em GUPTA (1997), o qual define *data warehouse* como um ambiente estruturado e extensível projetado para o trabalho de análise de dados não voláteis, lógica e fisicamente transformados oriundos de diferentes fontes e alinhados como os objetivos estratégicos da empresa.

Com o crescimento de produtos e serviços relacionados ao novo ambiente de *data warehouse*, surgiu o conceito de data warehousing como sendo um conjunto de tecnologias voltadas ao usuário responsável pela análise de conhecimento (executivo, gerente, analista) para melhorar o processo de tomada de decisões (CHAUDHURI & DAYAL, 1997).

Entretanto, um *data warehouse* pode ser usado para outras finalidades. Pela experiência obtida ao longo de anos em administração de banco de dados aprendemos que é de fundamental importância termos um banco de dados compacto. Neste sentido, o projeto de um ambiente de *data warehouse* é muito importante uma vez que permite que grandes volumes de dados (geralmente dados históricos) sejam transferidos do ambiente operacional para o ambiente de *data warehouse*. Isso traz uma série de vantagens ao ambiente operacional dentre as quais podemos citar:

- O banco de dados de produção torna-se mais fácil de:
    - Reestruturar; monitorar, corrigir e reorganizar.
- Redução do volume de dados no ambiente de produção e conseqüente ganho no desempenho das aplicações (menor tempo de resposta).

- Redução do tempo e mídia necessários para a realização do backup do banco de dados. Conseqüentemente, em caso de falha, a sua recuperação é mais rápida.

## 2.3. Características do data warehouse

O aspecto fundamental para a criação de uma *data warehouse* reside na separação dos dados. A separação dos dados do ambiente operacional para o ambiente de *data warehouse* possibilita um acesso mais efetivo aos dados para análise com o objetivo principal de, através da análise dos dados, obter vantagens competitivas.

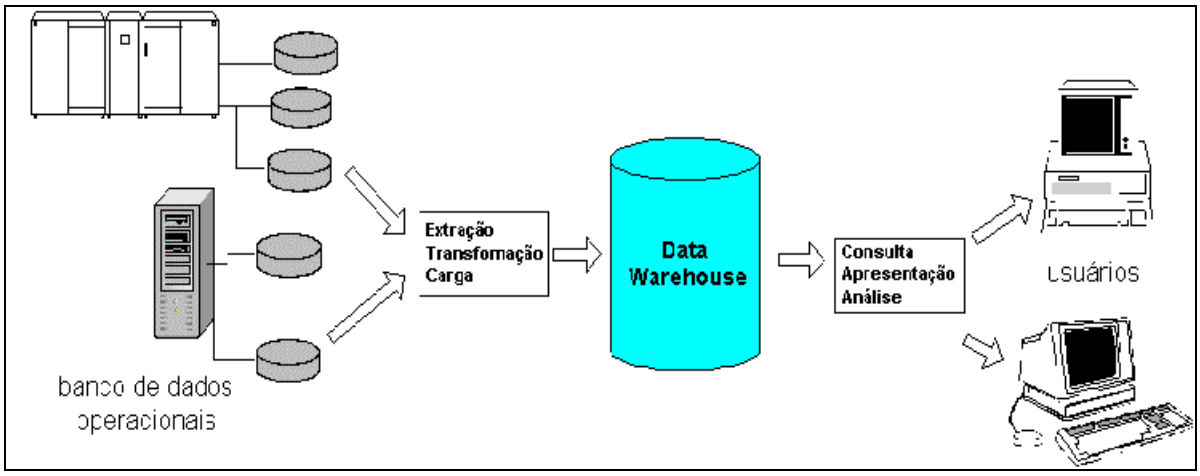



FIGURA 2.2: O ambiente de *data warehouse*

A figura 2.2 mostra um ambiente de *data warehouse* típico caracterizado pela separação do ambiente de dados operacional do ambiente de *data warehouse*. A seguir, enumeramos os aspectos mais importantes que caracterizam esse ambiente.

### 2.3.1. Integração dos dados

Muitas vezes para que o projeto de *data warehouse* tenha sucesso, ele precisa integrar dados oriundos de mais de um ambiente operacional. É bastante comum encontrar

empresas que possuem diferentes bancos de dados cada um atendendo a um sistema específico. No entanto, durante a fase de projeto do *data warehouse*, é natural que estes dados sejam integrados através da criação de um modelo único de dados a partir dos vários sistemas.

Praticamente todos os dados em um *data warehouse* são agrupados em função de um período de tempo. A maioria das atividades efetuadas em um *data warehouse* possui como critério de filtro um intervalo de tempo específico para análise.

À medida que os dados são derivados do ambiente operacional, passam por um processo de integração em que é preciso usar regras de conversão, ordenação e agregação, unificar entidades semelhantes, padronizar tamanhos, termos e valores etc. (BOHN,1997).

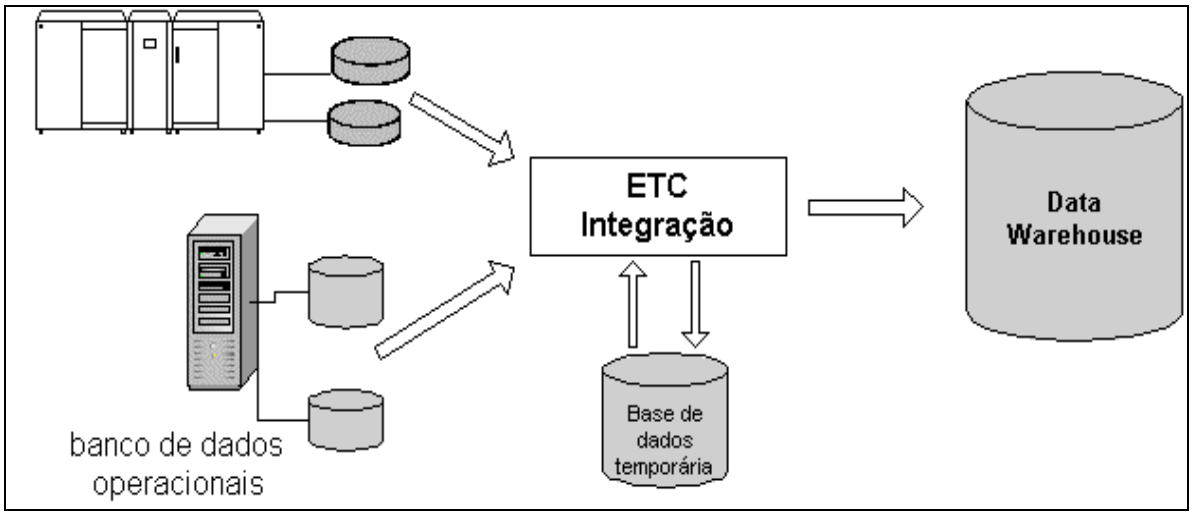



FIGURA 2.3: Processo de integração dos dados

Conforme mostra a figura 2.3, o processo de integração dos dados transforma os dados residentes nos bancos de dados operacionais transportando-os para o *data warehouse*. O processo de ETC (Extração, Transformação e Carga) geralmente é operacionalizado com o auxílio de uma ferramenta de integração de dados. O processo de transformação pode envolver os seguintes sub-processos:

**Limpeza dos dados:** Relacionados a erros de domínio de atributos e inconsistência dos dados de tabelas diversas.

**Conversão de tipos de dados:** Os dados oriundos de fontes diversas podem ter tipos diferentes e devem ser convertidos antes de passa-los ao *data warehouse*.

**Cálculos e derivações:** As regras são definidas na fase de levantamento de requisitos e aplicadas no momento da integração.

**Agregação:** Sumarização dos dados observando o nível de granularidade do *data warehouse*.

Durante o processo de integração dos dados, é muito comum a utilização de uma base de dados temporária cuja finalidade é armazenar as estruturas de dados criadas pelos processos de extração, transformação e carga dos dados.

### 2.3.2. Diferença entre processos transacionais e analíticos

Outra razão para separar o ambiente operacional do ambiente de *data warehouse* está relacionada ao fator performance. Os processos de análise típicos em um ambiente de *data warehouse* normalmente consomem grandes quantidades de recursos (Memória, CPU, Disco Magnético) do sistema. No ambiente operacional, a alta disponibilidade e performance são primordiais para que o sistema proporcione o tempo de resposta satisfatório (GUPTA, 1997).

Os ambientes operacionais são projetados para oferecer uma performance satisfatória. As transações que são executadas neste ambiente podem ser facilmente identificadas e os horários de picos são conhecidos. Desta forma, é possível efetuar um dimensionamento e gerenciamento de carga.

Em um ambiente de *data warehouse* os processos executados são mais complexos e muitas vezes acessam ou manipulam uma grande quantidade de dados. O processo de exploração dos dados, como a geração de resumos ou detalhamento, efetuado pelo usuário do *data warehouse* não possui um padrão definido. Conseqüentemente, o dimensionamento de carga fica bem mais difícil de ser efetuado. Os picos de uso nestes ambientes são completamente diferentes do ambiente operacional e não têm horários conhecidos. A tabela

2.1 enumera as principais diferenças de uma base de dados operacional de uma base de dados de *data warehouse* (PEREIRA, 1999)

| **Base de dados Operacionais** | ***Data warehouse*** |
|---|---|
| Objetivos operacionais | Registro histórico |
| Acessos leitura / escrita | Acessos de consulta |
| Transações acessam poucos registros | Muitos registros analisados |
| Estrutura normalizada | Existência de redundância |
| Modelagem relacional | Modelagem Dimensional |
| Consistência baseada em transação | Consistência com base no processo de alimentação |

TABELA 2.1: Base de dados operacional X *data warehouse*

### 2.3.3. Dados não voláteis

A maioria dos dados transportados para o *data warehouse* é não volátil, ou seja, os dados não sofrem alterações. No ambiente operacional, é comum ocorrer alterações nos dados em função da natureza do ambiente. Por exemplo, a alteração de endereço do cliente ou do número do telefone é uma operação corriqueira em um ambiente operacional. É importante perceber que uma vez feito o transporte dos dados para o ambiente de *data warehouse* ele dificilmente será alterado. É muito difícil, senão impossível estabelecer um sincronismo entre o ambiente operacional e o ambiente de *data warehouse*. O processo de análise dos dados no *data warehouse* exige que este ambiente seja estático dentro do período analisado.

O dado em um DATA WAREHOUSE refere-se a um momento específico no tempo. O tratamento de séries temporais adiciona complexidade ao ambiente de DW, pois a evolução da descrição do dado (metadado) torna difícil a agregação ao longo do tempo. Uma vez definida a unidade de tempo com que se quer armazenar dados em um DW (periodicidade), o mesmo passa a receber cópias em intervalos regulares identificando-os no tempo e disponibilizando-os para consulta (KLEIN, 1999)

### 2.3.4. Dados históricos

Em um ambiente operacional quando os dados passam para o status de inativos, é comum que sejam arquivados. A principal razão disso é tornar o banco de dados mais compacto e conseqüentemente obter melhor performance. Manter grandes quantidades de dados históricos misturados ao ambiente operacional pode diminuir o tempo de resposta das transações. Os *data warehouse*s são projetados para armazenar grandes quantidades de dados e por um período de tempo maior. O custo para manter o dado no *data warehouse* é mínimo. A maior parte dos custos está relacionada ao processo de transferência dos dados para o *data warehouse*.

Após a integração dos dados, estes são transportados para o *data warehouse* e armazenados para contemplar as necessidades de processamento analítico da empresa. Como estes dados não são mais atualizados, adquirem o caráter de somente leitura o que possibilita a existência de grandes volumes de dados históricos no *data warehouse*, sendo esta uma de suas principais características.

### 2.3.5. Acesso através de uma ferramenta front-end ou aplicação

Na maioria dos sistemas de *data warehouse*, o ambiente de acesso aos dados constitui uma outra camada na arquitetura do *data warehouse*. Este ambiente de acesso aos dados engloba as ferramentas *front-end*, aplicações, treinamento e suporte necessário para prover o acesso ao *data warehouse* e geração de informação para suporte a decisão. Em muitos casos, a tecnologia para acesso aos dados está baseada em um ambiente cliente/servidor. Uma estação é utilizada como cliente e o *data warehouse* como servidor. Entretanto, a medida mais *data warehouse*s vão sendo construídos, muitas variações no ambiente de acesso as dados são utilizados.

## 2.4. Razões para implementação de um data warehouse

Vamos discutir agora as principais razões que levam as empresas a implementar um ambiente de *data warehouse*. A maioria da literatura de *data warehouse* existente é um pouco confusa neste aspecto. Por exemplo, após um certo tempo gasto na leitura de livros e artigos sobre o assunto, encontramos afirmações tais como: “Obter vantagem competitiva”, “Converter dados em *Business Intelligence*” e “aproximar-se dos clientes”. Entretanto, a construção do *data warehouse* representa apenas um passo (embora complexo) no longo caminho em direção a estes objetivos maiores citados acima (GREENFIELD, 2002).

### 2.4.1. Separação de ambientes

Separar os processos de consulta e relatórios que são grandes consumidores de disco/memória dos sistemas transacionais:

A maioria das empresas configura seus sistemas operacionais com o objetivo de garantir um tempo de resposta satisfatório. Os complexos processos de consulta e relatórios comuns nos sistemas de processamento analítico tornam difícil de configurar um sistema para que tenha um tempo de resposta aceitável. A implementação de uma arquitetura de *data warehouse* é uma forma de separar os ambientes transacional e analítico.

### 2.4.2. Desempenho

Uso de modelos de dados e tecnologias que aceleram as consultas e relatórios. Existem maneiras de modelar que geralmente melhoram os processos de consulta e geração de relatórios e que não são apropriadas para os sistemas de processamento de transações pois degradam o desempenho geral destes sistemas. Além disso, existem tecnologias para servidores que podem melhorar a performance das consultas e relatórios mas que podem diminuir o desempenho dos sistemas transacionais (ex. índices bit-mapped). É importante

ressaltar que o quanto uma tecnologia melhora/prejudica o desempenho varia do produto usado e de como a tecnologia é usada.

Em MADEIRA (2001) são abordados alguns aspectos que influenciam no desempenho de uma *data warehouse*:

- Bom projeto lógico dos esquemas estrela
- Aspectos físicos do banco de dados
- Ajustes no *hardware*
- Ajustes no *software* (Sistema operacional e SGBD)
- Uso de agregados (materialização de visões)
- Processamento paralelo e particionamento

### 2.4.3. Propósito de uso

Criar um ambiente onde há necessidade de relativamente pouca experiência em tecnologia de banco de dados. Muitas vezes, um *data warehouse* é projetado para atender a processos simples de consultas e relatórios que podem ser elaborados por pessoas que não tenham qualificação avançada em informática.

Prover um ambiente que tenha dados de um período maior do que aqueles mantidos pelos ambientes operacionais. Dados históricos são muitas vezes removidos dos sistemas operacionais para que a performance deste ambiente possa ser melhor gerenciado. Estes dados removidos podem ser armazenados em um *data warehouse* que é um ambiente que necessita de um grau menor de gerência e controle de tempo de resposta. Uma vez inseridos no *data warehouse* estes dados podem permanecer por um período maior de tempo.

Prevenir que pessoas que apenas consultam dados de acessar os ambientes transacionais. Pode ser muito útil criar uma base de dados que atenda as consultas feitas pela internet. Neste ambiente o aspecto segurança é crítico. Neste sentido um *data warehouse* para atender este propósito é uma solução adequada.

## 2.5. Arquitetura de data warehouse

A escolha da arquitetura é uma decisão gerencial baseada principalmente na infraestrutura atual existente, ambiente de desenvolvimento, escopo de implementação, disponibilidade de recursos financeiros e capacidade técnica (pessoas) da empresa. Ainda que a escolha da arquitetura possa ser retardada, é fundamental que ela seja definida no inicio do projeto. A mudança na arquitetura durante o projeto pode exigir que parte do trabalho seja refeita.

A escolha do tipo de implementação do *data warehouse* também é uma decisão gerencial que é baseada principalmente na arquitetura de *data warehouse* escolhida, tempo necessário para implementação, satisfação do usuário, recursos necessários em todas as fases do projeto e retorno do investimento.

Uma arquitetura é um conjunto de normas que proporcionam uma estrutura para o projeto de um sistema ou produto. A arquitetura proporciona uma visão da estrutura do *data warehouse* e auxilia no entendimento de como ocorre o fluxo dos dados ao longo do processo de data warehousing. O requisito fundamental do *data warehouse* é ter flexibilidade para suportar a dinamicidade das necessidades de análise dos dados. Para atender a este requisito a arquitetura deve incluir os aspectos-chave relacionados aos três principais componentes de *data warehouse*:

População do *warehouse*: Passagem dos dados do ambiente operacional para o *data warehouse*. Esta é a etapa que compreende o maior volume de trabalho pois engloba as tarefas de extração, limpeza e transformação dos dados.

Administração do *warehouse*: Manutenção dos metadados. Os metadados fornecem informações sobre os relacionamentos dos dados armazenados no *data warehouse*.

Ferramentas para acesso e análise: Os usuários acessam o *data warehouse* através de uma ferramenta de análise e exploração dos dados. Os sistemas aplicativos construídos especificamente para o *data warehouse* também constituem parte do conjunto de ferramentas utilizadas para produção de informações para suporte a decisão.

A escolha da arquitetura de *data warehouse* determina, ou será determinada pela localização do *data warehouse* empresarial e dos *data warehouse*s departamentais e de onde eles serão administrados. Um *data warehouse* pode ser implementado e administrado de forma centralizada ou distribuída ou uma combinação de ambos. Em seu livro, POE et al, (1998) apresenta uma arquitetura genérica e a partir dela analisa duas variações desta arquitetura conforme descrito abaixo.

### 2.5.1. Arquitetura de data warehouse genérica

A figura 2.4 mostra a arquitetura genérica de um *data warehouse*. Tal arquitetura também conhecida como arquitetura global. Neste modelo, os dados são extraídos dos sistemas fonte e integrados. Após o processo de integração, os dados são transportados para o banco de dados do *data warehouse*. Os usuários acessam os dados do *data warehouse* através de uma uma ferramenta de consulta que possui as funcionalidades de transformar estes dados em informação útil.

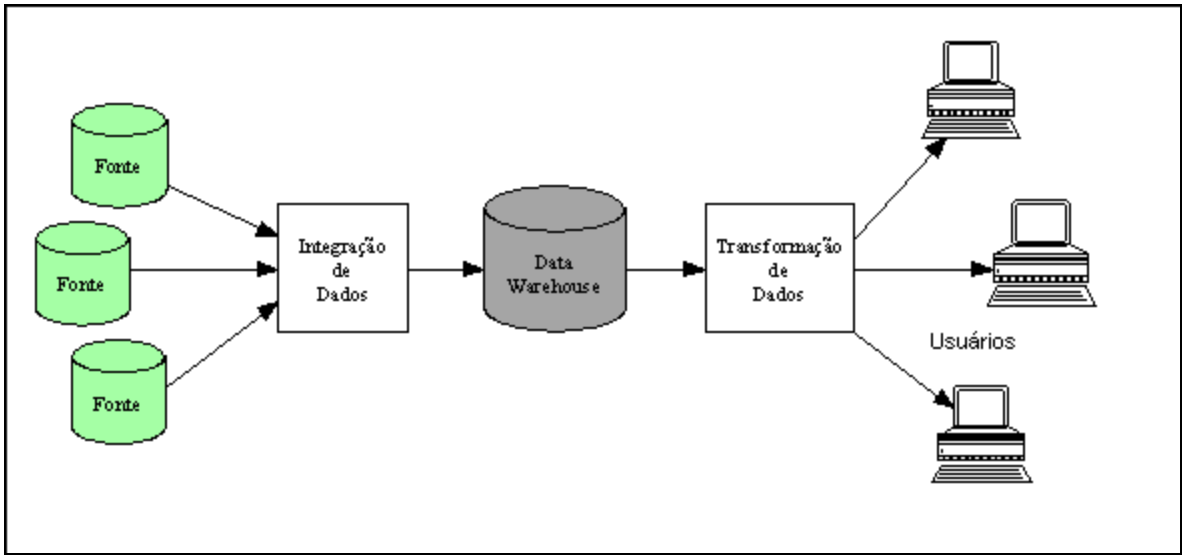



FIGURA 2.4: Arquitetura Genérica de um *Data warehouse*

A arquitetura global de um *data warehouse* possui suas próprias características que o diferencia de outros sistemas. Estas características foram descritas no Capítulo 2 – Ambiente de *Data warehouse*. Parte do trabalho na construção do *data warehouse* é incorporar os aspectos principais que caracterizam a construção de um *data warehouse* genérico na arquitetura atual para atender as necessidades de processamento de suporte a decisão. É importante entender que o *data warehouse* genérico pode ser implementado de várias maneiras diferentes com graus diferenciados de sofisticação.

## 2.5.2. Expansão da arquitetura de data warehouse genérica

A partir do *data warehouse* genérico, as empresas podem implementar variações deste modelo. A Figura 2.5 mostra um *data warehouse* corporativo alimentando os *data data warehouses* departamentais (tambéma chamados de *data mart*s).

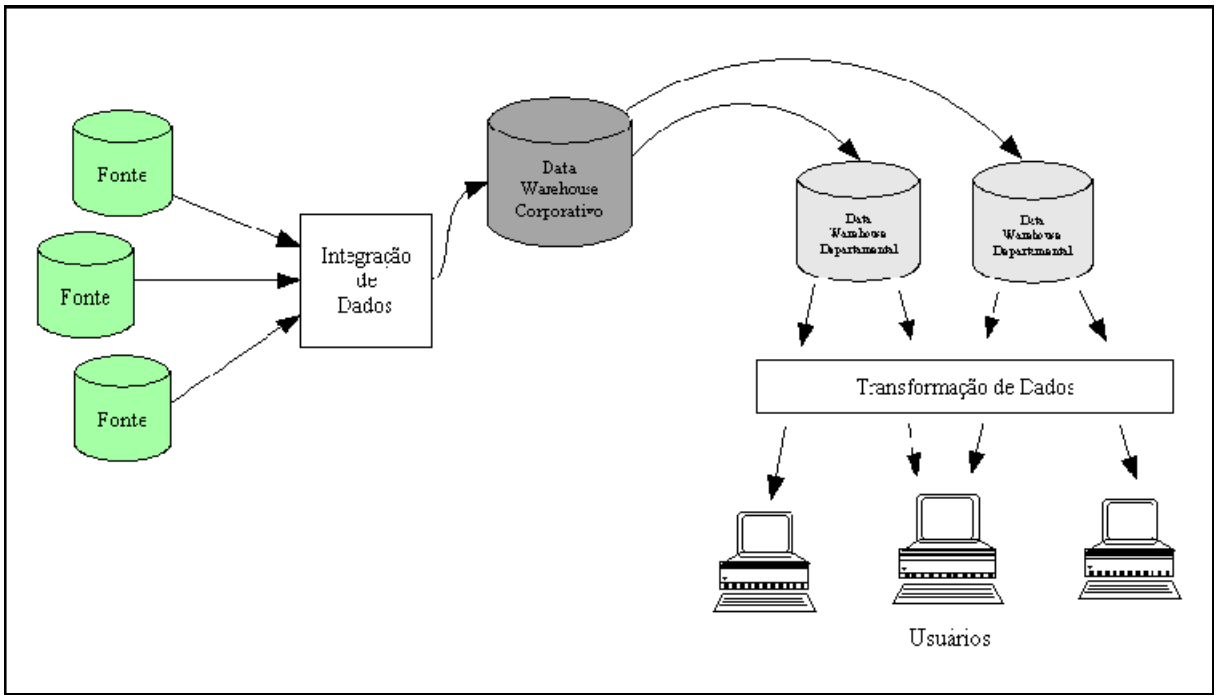



FIGURA 2.5: *Data warehouse* Corporativo alimentando *data mart*s

Qual a justificativa para uma empresa usar esta arquitetura ? Existem várias razões para isso:

- Para atender uma estratégia da empresa a qual usa um modelo de dados empresarial como base;
- Para separar o processo de integração dos dados dos processos de projeto de banco de dados e desnormalização;
- Para usar o *data warehouse* empresarial como fonte consistente para todos os departamentos da empresa.

A integração de uma arquitetura de *data warehouse* com a arquitetura do sistema de processamento atual nem sempre é direta e simples como pode parecer. As arquiteturas dos sistemas de processamento em muitas empresas são bastante sofisticadas. Na maioria das empresas existe algum tipo de obstáculo na implementação de uma arquitetura. Estes obstáculos podem ser de natureza técnica, estratégica, integração ou considerações políticas que impõem limitações de como a arquitetura de *data warehouse* será implementada na empresa.

Como uma arquitetura é um conjunto de regras ou estruturas que proporcionam um esqueleto para o projeto de um produto ou sistema completo ela não pode ser separada do *data warehouse* propriamente dito. A implementação de uma arquitetura de *data warehouse* pode ser feita de várias maneiras e podem tornar-se bastante complicadas em uma empresa. Entretanto, o esqueleto básico do *data warehouse* deve ser implementado. Não há uma maneira correta de implementar uma arquitetura de *data warehouse*. Além disso, existem diversas soluções técnicas possíveis para atender aos diversos ambientes empresariais e suas necessidades específicas.

No exemplo da figura 2.5, o esqueleto básico da arquitetura genérica de *data warehouse* está sendo usado. Os dados são extraídos dos sistemas fonte. Após a extração, os dados são integrados antes de serem carregados para dentro do *data warehouse* empresarial. Estes dados são então reestruturados, re-projetados e carregados em *data warehouse*s departamentais. O usuário acessa o *data warehouse* específico do seu departamento através de uma ferramenta apropriada. Esse é de fato uma arquitetura de *data*

*warehouse* que foi modificada para atender a objetivos e necessidades de suporte à decisão específicos de cada departamento.

Parte da construção de um *data warehouse* é o processo de descobrir a solução técnica mais adequada para as necessidade de informações de suporte a decisão e criar uma arquitetura de *data warehouse* sólida dentro dos parâmetros que devem ser respeitados. Uma arquitetura não é melhor que outra. Uma arquitetura pode ser mais difícil e consumir mais tempo para ser desenvolvida que outra. Na maioria dos casos, entretanto, a arquitetura de *data warehouse* escolhida é aquela que representa a solução mais apropriada para os requisitos técnicos, objetivos da empresa e necessidades de suporte a decisão.

### 2.5.3. Data warehouse organizado por assunto

Outra implementação de arquitetura de *data warehouse* é mostrada na figura 2.6

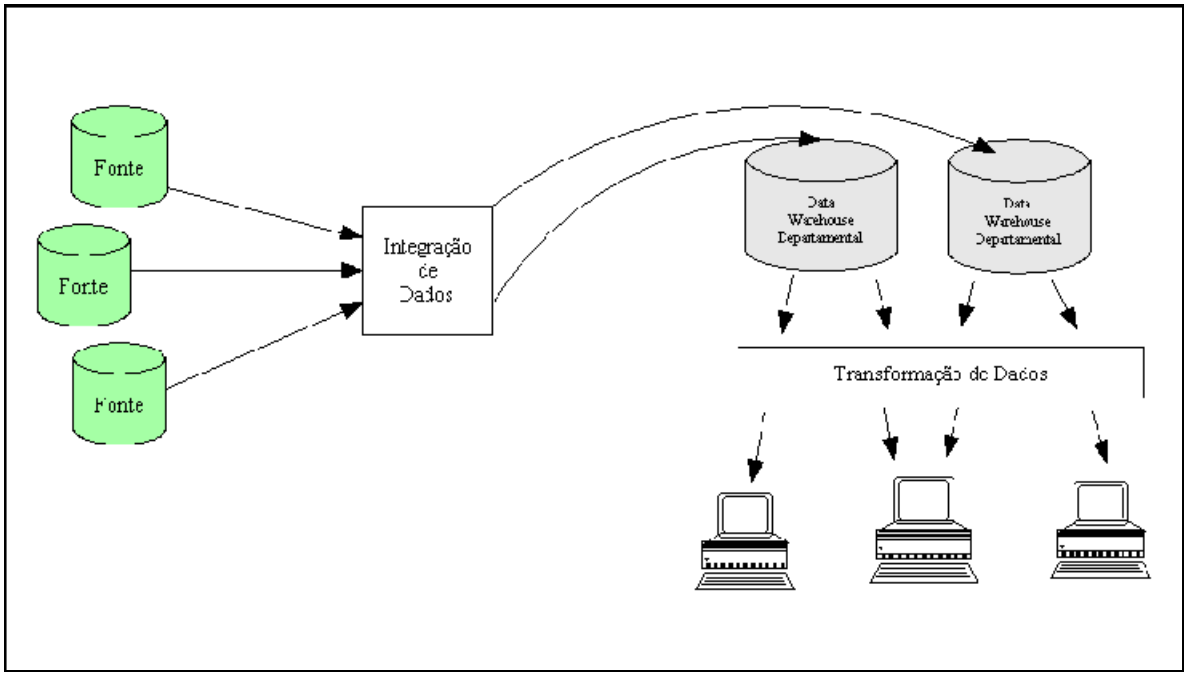



FIGURA 2.6: *Data warehouse*s departamentais

Neste modelo, os dados passam pelo processo de integração e então carregados em *data warehouse*s departamentais. Estes *data warehouse*s departamentais também são conhecidos como *data mart*s e são controlados pelo próprio departamento pois são construídos com a finalidade de atender as necessidades específicas de cada área da empresa. O fato de não existir um banco de dados a nível empresarial não impede que este seja um verdadeiro *data warehouse*. É claro que poderá haver limitação quanto ao tamanho do banco de dados na implementação desta arquitetura. Um *data mart* pode estar interconectado com outro *data mart* de outro departamento. O departamento de informática auxilia na construção dos *data mart*s pois é ele quem detém o conhecimento da estrutura dos bancos de dados operacionais e pode auxiliar no processo de extração dos dados. Geralmente os departamentos onde o *data mart* é implementado não possui recursos e habilidades técnicas suficientes para implementar o *data mart*. O departamento de informática gerencia o processo. Quem determina o conteúdo do *data mart* é o departamento que fará uso dele para atender as suas necessidades de informação.

O fato de criar *data warehouses* departamentais pode ocasionar um grande grau de redundância de dados. Á medida que cresce o número de departamentos implementando seu *data warehouse*, cresce também o grau de redundância uma vez que não há qualquer compartilhamento de dados entre os departamentos.

## 2.6. Infraestrutura de Data warehouse

A infraestrutura está estreitamente relacionada com a arquitetura de *data warehouse*. A infraestrutura corresponte aos componentes necessários para implementar determinada arquitetura. POE et al. (1998) classifica os componentes da infraestrutura em três grandes grupos: Hardware, software e treinamento (Qualificação).

A Figura 2.7 enumera um conjunto de possíveis componentes integrantes da infraestrutura para a implementação de uma arquitetura de *data warehouse* genérica. Cabe destacar que para uma mesma arquitetura, diferentes infraestruturas podem ser necessárias

considerando as especificidades relativas ao ambiente da organização onde o *data warehouse* será implementado.

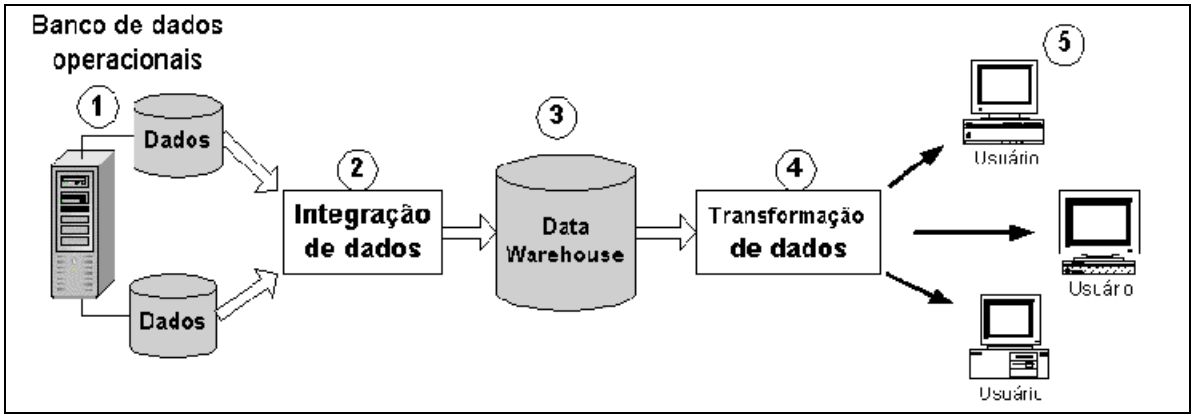



FIGURA 2.7. Possíveis componentes da infraestrutura de *data warehouse*

A título de exemplo, vamos assumir que o ambiente representado pela figura 2.7 seja a arquitetura escolhida por uma universidade que pretende construir o seu primeiro *data warehouse.* Os dados fontes estão armazenados em bancos de dados relacionais de fornecedores diferentes. O *data warehouse* resultante será composto pela integração dos dados originados de ambos os bancos de dados. Os usuários irão acessar os dados do *data warehouse* através de uma ferramenta de análise e exploração de dados. Dessa forma, os componentes da infraestrutura poderiam ser:

1. Treinamento da equipe envolvendo os conceitos e tecnologias de *data warehouse.* Este treinamento é considerado como parte da infraestrutura uma vez que a equipe está iniciando o primeiro projeto de *data warehouse.*
2. Ferramentas para extração, transformação e carga dos dados dos ambientes operacionais para o *data warehouse.*
3. Habilidades de administração de banco de dados (DBA).
4. Ferramentas para o controle, gerenciamento dos metadados.
5. Instalação, configuração e treinamento dos produtos necessários para a infraestrutura de rede, comunicações e estações de trabalho.

A escolha da infraestrutura constitui um aspecto importante para o sucesso de um projeto de *data warehouse* bem como representa um percentual significativo do montante de custos envolvidos no projeto (POE, 1998).

## 2.7. Opções de implementação

Existem basicamente duas abordagens para implementar a arquitetura selecionada. *Top Down* e *Bottom up*. A partir delas, muitas variações são possíveis. Em alguns casos, uma combinação de ambas pode ser utilizada. Vários fatores influenciam ou mesmo determinam a escolha da abordagem de implementação:

- Infraestrutura do departamento de informática
- Recursos disponíveis
- Tecnologias necessárias
- Escopo da implementação
- Arquitetura selecionada
- Tempo para implementação
- Necessidade de acesso a dados globais da empresa

### 2.7.1. Implementação top down

A implementação t*op down* requer que os trabalhos de planejamento e projeto sejam efetuados no início do projeto. Isso traz consigo a necessidade de envolver pessoas de todos os departamentos envolvidos na implementação do *data warehouse*. Antes mesmo de iniciar a implementação do *data warehouse* deverão ser tomadas decisões sobre fontes de dados e sua estrutura, segurança, padronização e um modelo de dados genérico. De uma forma geral, quando for selecionada a implementação top down uma arquitetura *data warehouse* global é implementada. A construção de *data mart*s individuais pode ser feita a

partir do *data warehouse* global ao invés de serem populados diretamente do bancos de dados operacionais.

Esta abordagem de implementação pode resultar numa definição de dados mais consistente. Entretanto o custo inicial de análise e projeto pode ser significativo. Esta etapa consome grande volume de tempo e pode retardar a implementação e os benefícios que o *data warehouse* poderia representar.

A abordagem *Top Down* pode ser uma boa escolha para as empresas que possuem um departamento de informática centralizado que gerencia todos os recursos de *hardware* e *software*. Neste caso, os demais departamentos não possuem os recursos necessários para implementar seus próprios *data mart*s. Nas empresas onde os departamentos possuem seus próprios recursos de *hardware* e *software* esta opção pode não ser adequada. Dificilmente haverá concordância em esperar para definir e implementar uma infraestrutura global.

### 2.7.2. Implementação bottom up

A implementação b*ottom up* permite que o trabalho de planejamento e projeto dos *data mart*s departamentais sejam efetuados sem que haja uma infraestrutura global. Isto não significa que a infraestrutura global não seja desenvolvida. A infraestrutura global vai sendo construída de forma incremental na medida em que os *data mart*s vão sendo criados. Esta abordagem é mais amplamente aceita comparada com a abordagem *top down* pois permite que os resultados sejam apresentados e usados para justificar e expandir o modelo e viabilizar uma implementação global. Cada *data mart* criado será adicionado ao *data warehouse* global. Inicialmente a população dos *data mart*s é feita diretamente dos bancos de dados operacionais. A medida que o *data warehouse* cresce através da incorporação dos *data mart*s, novos *data mart*s especializados poder surgir. Estes por sua vez podem ser populados a partir do *data warehouse*.

A implementação b*ottom up* inicial geralmente é mais barata considerando que menos recursos de *hardware* e humanos serão necessários. Esta tem sido a escolha da maioria das empresas para a implementação de *data warehouse*. A causa principal desta escolha é maior rapidez de retorno do investimento.

A desvantagem principal desta abordagem é a possibilidade de redundância e inconsistência dos dados. a medida que mais *data mart*s são criados. Com planejamento e monitoramento cuidadoso este problema pode ser minimizado. A integração dos *data mart*s para um ambiente de *data warehouse* global pode ser difícil a menos que algum trabalho de planejamento tenho sido feito.

### 2.7.3. Implementação mista

Ambas as abordagens vistas oferecem vantagens e desvantagens. Em muitos casos, a implementação mista (uma combinação da abordagem *top down* e *bottom up*) pode ser a melhor escolha. Neste caso um gerente qualificado pode ser necessário para efetuar o balanceamento desta combinação. O ponto principal nesta abordagem é a determinação do grau de planejamento e projeto que é necessário para suportar a integração futura dos *data warehouse* construídos através da abordagem *bottom up*. A partir disso, desenvolver uma infraestrutura básica para o *data warehouse* global como a preocupação inicial de abranger somente a áreo do negócio sendo abordado.

A medida que os *data mart*s são implementados é recomendado desenvolver um plano para tratar os elementos de dados que são utilizados em mais de um *data mart*. Em algumas circunstâncias a redundância dos dados no *data mart* pode ser apropriado. Neste caso deve-se avaliar o custo benefício entre capacidade de armazenamento, facilidade de acesso e o impacto causado pelo trabalho adicional de manter a redundância no nível aceitável de consistência.

Na implementação de um *data warehouse* surgem muitos pontos que precisam ser resolvidos. Através da abordagem mista, estes podem ser resolvidos na medida em que estes vão surgindo, dentro de um escopo menor que é o *data mart* comparado ao *data warehouse* global. O monitoramento cuidadoso do processo de implementação e o gerenciamento dos aspectos importantes podem resultar num ganho significativo através do uso do melhor que cada abordagem oferece.

## 2.8. Considerações finais

Neste capítulo apresentamos os principais aspectos que contribuíram para a evolução dos sistemas de apoio à decisão que originando os sistemas de *data warehouse*. Abordamos as características, arquitetura básica de um *data warehouse* e as razões da separação do ambiente de banco de dados operacional da organização do ambiente de banco de dados analítco.

O próximo capítulo aborda a modelagem de dados para *data warehouse* destacando a modelagem dimensional como alternativa a tradicional modelagem Entidade Relacionamento.

# 3. MODELAGEM DE DADOS PARA DATA WAREHOUSE

Este capítulo aborda o tópico relacionado à modelagem de dados. No ambiente operacional, a técnica de modelagem ER (Entidade Relacionamento) tem sido comumente utilizada. Com o advento dos sistemas e processos *data warehousing* surgiu a necessidade de usar uma técnica de modelagem de dados mais adequada para este ambiente. Neste capítulo, será apresentada a técnica de modelagem dimensional proposta por KIMBALL (1998).

## 3.1. O modelo de dados relacional

O modelo relacional, de substancial base teórica, substituiu com grande vantagem os modelos de rede e hierárquico (KORTH & SILBERSCHATZ, 1993), tornando-se o mais utilizado em sistemas OLTP. Essa teoria, baseada na álgebra relacional, auxilia o projeto de bancos de dados e o processamento eficiente de consultas por seus usuários.

O modelo ER, de indiscutível utilidade nos sistemas OLTP, parece não ser adequado às necessidades de aplicações sobre os negócios. KIMBALL (1998) cita razões que justificam essa afirmação:

- O modelo não é simétrico, ou seja, todas as entidades parecem iguais;
- Geralmente existem muitas ligações entre duas entidades do modelo, o que torna o processo de consulta confuso;

- Qualquer consulta, envolvendo várias tabelas mapeadas do modelo conceitual, com milhões de linhas, não oferecerá resultados concretos. Os diagramas são muito complexos, tanto para o entendimento do usuário, como para a navegação do *software* de consulta.

A decisão de usar a técnica de modelagem de dados tradicional (ER) ou a modelagem dimensional deve ser tomada preferencialmente antes do início da etapa de modelagem de dados e do projeto físico do banco de dados. Em ambos os casos a performance pode ser otimizada através da desnormalização e o particionamento dos dados. Entretanto, a modelagem dimensional é muitas vezes usada na modelagem do banco de dados para *data warehouse* pois (POE et al., 1998):

- Cria um projeto de banco de dados com tempo de resposta rápido.
- Permite que os otimizadores de banco de dados trabalhem com um projeto de banco de dados mais simples obtendo melhores planos de execução.
- Usuários estão acostumados a pensar e usar os dados desta forma.
- Simplifica o entendimento e navegação dos metadados para os desenvolvedores e usuários.
- Aumenta as opções para escolha da ferramenta de acesso aos dados uma vez que algumas ferramentas requerem o uso do modelo esquema estrela.

## 3.2. Modelagem dimensional

Ao contrário do modelo entidade/relacionamento, o modelo dimensional é muito assimétrico. Há uma tabela dominante no centro do diagrama com múltiplas junções conectando-a as outras tabelas. Cada uma das tabelas secundárias possui apenas uma junção com a tabela central. A tabela central é chamada de **tabela de fatos**. As tabelas secundárias são chamadas de **tabelas de dimensão** (KIMBALL, 1998). A figura 3.1 mostra um modelo dimensional o qual possui a tabela fato (VESTIBULAR) no centro do diagrama. As dimensões do fato (período, campus e curso) aparecem ao redor da tabela central.

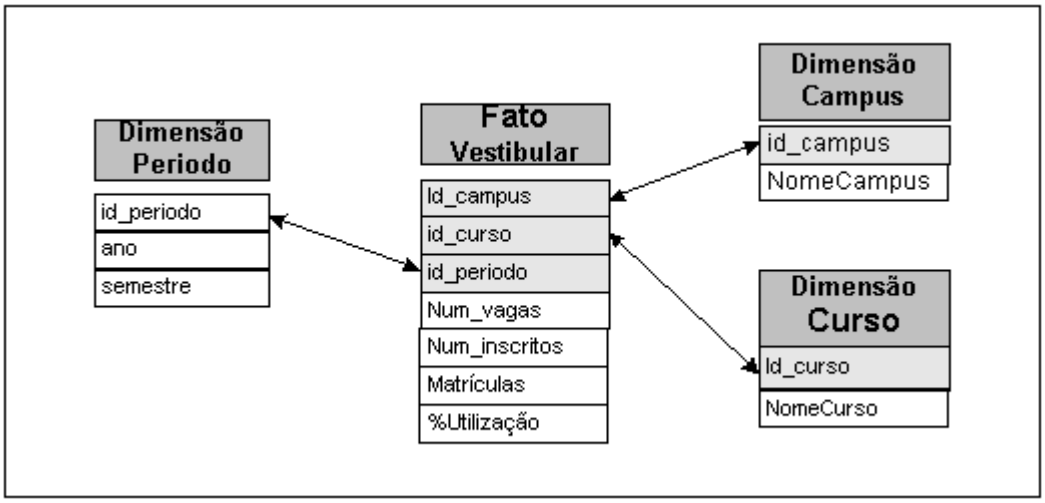



FIGURA 3.1: Um modelo dimensional típico

A modelagem dimensional é uma técnica para conceitualização e visualização de modelos de dados como um conjunto de medidas que são descritas pelos aspectos comuns do negócio. Esta técnica é especialmente útil para a sumarização e reorganização de dados para posterior apresentação ao usuário. Apesar de não ter firmeza em seus detalhes e técnicas de desenvolvimento por ser uma técnica relativamente nova  ela está sendo muito utilizada na construção de *data warehouse*s.

Um modelo dimensional tem três elementos básicos: (MACHADO, 2000)

- Fatos
- Dimensões
- Medidas (Variáveis)

### 3.2.1. Fatos

Um fato é uma coleção de itens de dados. O fato é omposto de dados de medidas e de dados de contexto. Cada ocorrência do fato representa um item de negócio, uma

transação de negócio ou um evento do negócio e é utilizado para analisar o processo das operações de uma empresa.

A característica básica de um fato é que ele é representado por valores numéricos e implementado em tabelas denominadas tabelas de fatos. Outra característica para identificar um fato é sua natureza evolutiva, mudando as suas medidas no tempo podendo ser questionadas. Consideremos o seguinte fato como exemplo: O percentual de utilização das vagas oferecidas no vestibular de 1997 e 1998 vem aumentando. Este fato está representado na tabela 3.1.

| | 1997 | | | 1998 | | |
|---|---|---|---|---|---|---|
| CURSO | Vagas | Matrículas | %Util. | Vagas | Matrículas | %Util. |
| ADM | 50 | 45 | 90% | 50 | 47 | 94% |
| CON | 45 | 41 | 91% | 50 | 48 | 96% |
| INF | 60 | 50 | 83% | 40 | 35 | 87% |
| DIN | 70 | 67 | 95% | 60 | 60 | 100% |

TABELA 3.1: Percentual de utilização das vagas

Esta afirmação foi realizada a partir da análise dos dados no período em questão. Porque consideramos *percentual de utilização das vagas* como um fato ? O percentual de utilização das vagas é o resultado de uma operação algébrica entre o número de vagas oferecidas sobre a quantidade de alunos matriculados. Então isso é um fato, pois para descrevê-lo usamos valores numéricos. Estes valores mudam ao longo do tempo.

Sempre que quisermos entender alguma coisa relativa a dados, informações, é recomendado que se formate uma tabela com esses valores para auxiliar no entendimento (MACHADO, 2000). O histórico dos fatos pode ser mantido e aumenta como o passar do tempo. Esse atributo também é uma característica que auxilia na identificação de um fato.

### 3.2.2. Dimensões

As tabelas de dimensão armazenam as descrições textuais das dimensões do negócio. Cada uma dessas descrições textuais ajuda a definir um componente da respectiva dimensão. Uma das principais funções dos atributos de tabelas de dimensão é servir como fonte para restrições em uma consulta ou como cabeçalhos de linha no conjunto de resposta do usuário.

Na maioria das vezes as dimensões representam hierarquias, como por exemplo, um produto, que é de uma marca ou categoria, que por sua vez pertence a uma sub-categoria etc. Só que, na maioria das vezes, quando esta é representada na dimensão, não temos várias tabelas normalizadas com ligações um-para-muitos, e sim uma única tabela de dimensão. Isso faz com que o desempenho das consultas aumente muito, já que não são necessários junções para se obter os dados relacionados com algum assunto (KIMBALL, 1996).

A tabela de fatos representa os relacionamentos muitos-para-muitos (m : n) entre as tabelas de dimensões, tendo como chave primária uma chave composta de todas as chaves estrangeiras das respectivas tabelas de dimensão (KIMBALL, 1997).

Existem muitas variações do modelo estrela, todas compartilhando do mesmo conceito de tabelas de fato e tabelas de dimensão associadas. Pode acontecer, por exemplo, de o negócio necessitar de múltiplas tabelas de fato para ser analisado. Quando isto acontece, temos um conjunto de esquemas estrela inter-relacionados, chamado galáxia (McGUFF, 1996).

Quando as dimensões são grandes, com alta cardinalidade, faz sentido usar o esquema floco de neve (*snowflake*), que separa as tabelas de dimensões em termos de atributos, diminuindo o grau de redundância de dados, como mostrado na Figura 3.2.

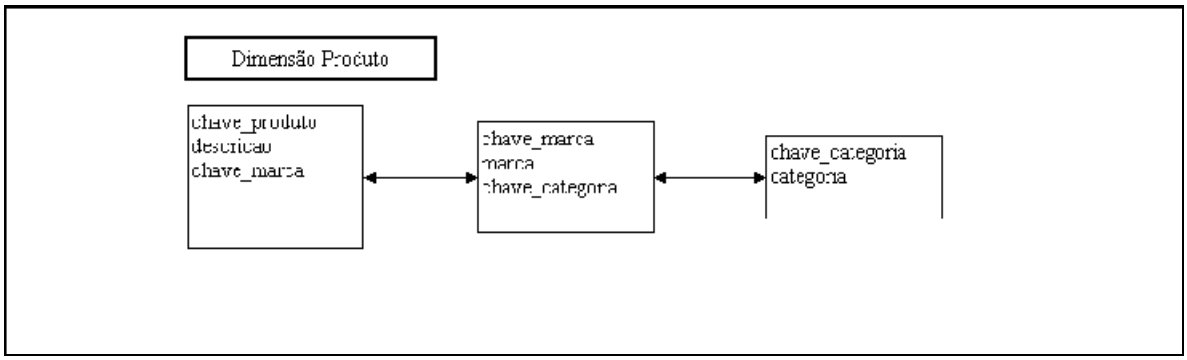



FIGURA 3.2: Um Esquema Floco de Neve

Segundo KIMBALL (1997), a modelagem multidimensional baseada no modelo estrela apresenta várias vantagens no uso em DW, dentre estas estão:

- O modelo estrela tem uma arquitetura padrão e previsível. As ferramentas de consulta e interfaces do usuário podem se valer disso para fazer suas interfaces mais amigáveis e o processamento mais eficiente;
- Todas as dimensões do modelo são equivalentes, ou seja, podem ser vistas como pontos de entrada simétricos para a tabela de fatos;
- Todas as tabelas de fato e dimensões podem ser alteradas simplesmente acrescentando novas colunas às tabelas;
- Nenhuma ferramenta de consulta ou relatório precisa ser alterada de forma a acomodar as mudanças;
- Todas as aplicações que existiam antes das mudanças continuam rodando sem problemas;
- Produtos heterogêneos: quando um negócio, tal como um banco, precisa controlar diferentes linhas de negócio juntas, dentro de um conjunto comum de atributos e fatos, mas ao mesmo tempo esta precisa descrever e medir as linhas individuais de negócio usando medidas incompatíveis; e
- Um número cada vez maior de utilitários administrativos e processo de *software* serem capazes de gerenciar e usar agregados, que são de suma importância para a boa performance de respostas em um DW.

### 3.2.3. Medidas

Uma medida é um atributo numérico de um fato o qual representa o desempenho ou o comportamento do negócio relativo as dimensões que participam desse fato. Esses números são denominados de variáveis. Como exemplos de medidas temos: *Valor total de vendas, número de unidades compradas, numero de unidades vendidas, numero de unidades devolvidas, custo dos produtos produzidos*. A medida é determinada pela combinação das dimensões que participam de um fato e estão localizadas como atributos na tabela de fatos.

### 3.2.4. Representação do modelo dimensional

Enquanto que na estrutura relacional os dados são apresentados de forma tabular conforme mostra a Tabela 3.2, a estrutura multidimensional proporciona uma visão matricial com um número fixo de dimensões e os valores são armazenados nas células da matriz. Cada dimensão possui um número fixo de elementos.

| CAMPUS | CURSO | VAGAS |
|---|---|---|
| C1 | Administração | 50 |
| C1 | Informática | 40 |
| C1 | Pedagogia | 30 |
| C2 | Administração | 30 |
| C2 | Informática | 50 |
| C2 | Pedagogia | 50 |
| C3 | Administração | 40 |
| C3 | Informática | 30 |
| C3 | Pedagogia | 20 |

TABELA 3.2: Visão tabular do ambiente relacional

A forma de organização dos dados derivada da análise dimensional gera a idéia de um cubo, apesar de muitas vezes dificultar o entendimento pela complexidade inerente quando da utilização de várias dimensões. Esta complexidade diminui à medida que inicia-se uma análise mais detalhada do cubo através da divisão do mesmo em fatias menores para facilitar o entendimento. O cubo é, de fato, apenas uma metáfora visual. É uma representação intuitiva do evento porque todas as dimensões coexistem para todo ponto no cubo e são independentes umas das outras (CAMPOS & ROCHA, 2001). A Figura 3.3 apresenta a visão matricial a qual facilita o entendimento e visualização de situações típicas em um ambiente de suporte à decisão.

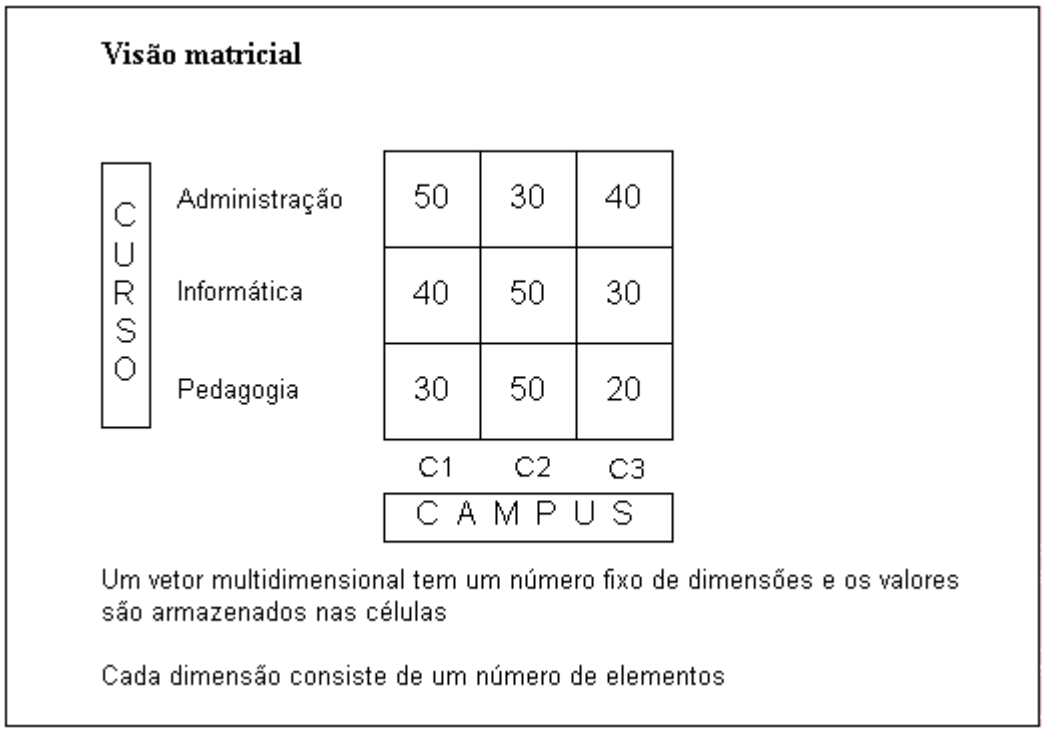



FIGURA 3.3 : Visão matricial

A forma mais popular de representação do modelo dimensional é o desenho de um cubo. Usualmente, um modelo dimensional consiste de mais de três dimensões sendo então representado através de hipercubo. Visualizar graficamente um hipercubo é muito difícil. Por esta razão, utiliza-se o cubo para representar qualquer modelo multidimensional.

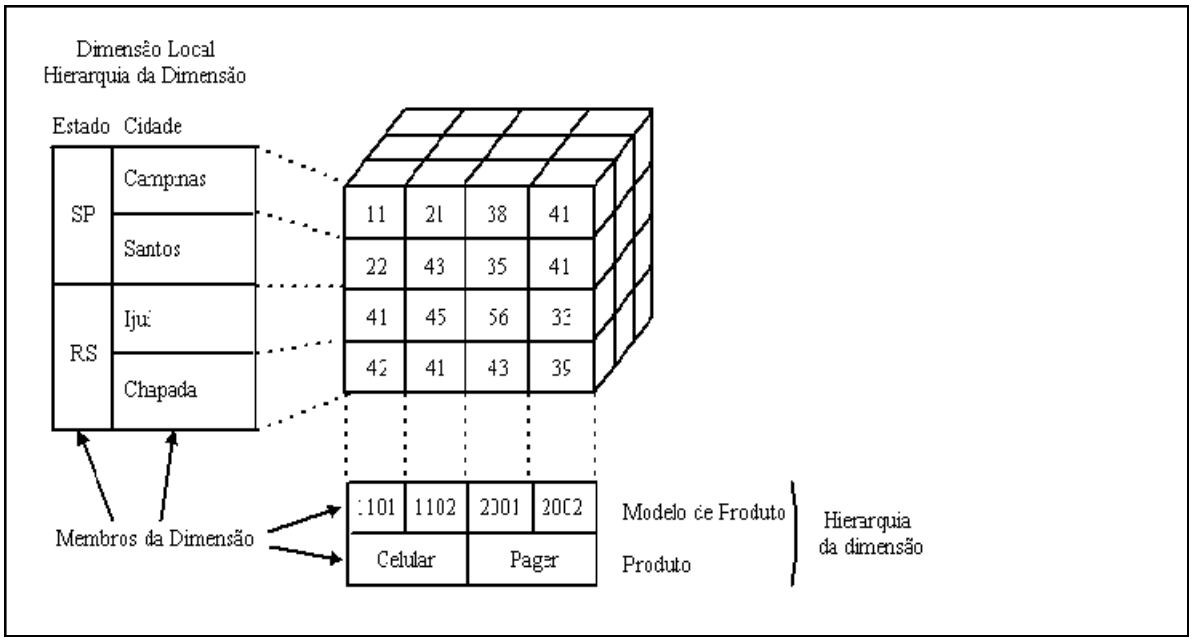



FIGURA 3.4: Representação do modelo dimensional

A Figura 3.4 representa o modelo dimensional o qual mostra o volume de produção que é determinado pela combinação de três dimensões: **local**, **produto** e **tempo**. As dimensões *local* e *produto* possuem dois níveis de hierarquia. A dimensão *local* está dividida em *cidade* e *estado*. A dimensão produto está dividida em *produto* e *modelo do produto*.

### 3.2.5. Operações básicas

A característica básica da modelagem dimensional é prover suporte ao processamento analítico e ao processo de tomada de decisões. Neste processo, quatro tipos de operações são utilizadas. Para visualizar os dados em um nível maior/menor de detalhe, são usadas as operações de *drill down* e *roll up*. Através das operações *slice and dice*, é possível visualizar os dados em diferentes perspectivas navegando pelas dimensões do modelo.

### 3.2.5.1. Drill down e roll up

Esta operação é usada para visualizar os dados ao longo dos níveis hierárquicos de uma dimensão. A tabela 3.3 mostra o efeito da operação *roll up* sobre a dimensão tempo.

| Volume de produção (em milhares) | | 1 9 9 9 | | | |
|---|---|---|---|---|---|
| | | Trimestre 1 | Trimestre 2 | Trimestre 3 | Trimestre 4 |
| Região Sul | RS | 78 | 67 | 22 | 56 |
| | SC | 90 | 67 | 88 | 99 |

***Roll up***
**Dimensão Tempo**

| Volume de produção (em milhares) | | Trimestre 1 | | |
|---|---|---|---|---|
| | | Janeiro | Fevereiro | Março |
| Região Sul | RS | 30 | 26 | 22 |
| | SC | 28 | 30 | 32 |

TABELA 3.3: Operação *Roll up* – Dimensão tempo

Com a capacidade de *drill down* o usuário pode navegar do mais alto nível de detalhe até o mais baixo nível de detalhe, expandindo sua visão na hierarquia da dimensão.

| Volume de produção (em milhares) | | Telefone Celular | | Pagers | |
|---|---|---|---|---|---|
| | | 1001 | 1002 | 2001 | 2002 |
| Região Sul | RS | 33 | 12 | 8 | 12 |
| | SC | 45 | 34 | 20 | 23 |

***Drill down***
**Dimensão local**

| Volume de produção (em milhares) | | Telefone Celular | | Pagers | |
|---|---|---|---|---|---|
| | | 1001 | 1002 | 2001 | 2002 |
| RS | Ijuí | 13 | 4 | 2 | 5 |
| | Chapada | 20 | 8 | 6 | 7 |

TABELA 3.4: Operação *Drill down* – Dimensão local

A tabela 3.4 ilustra a operação *Drill down* sobre a dimensão local. Com esta operação, o usuário pode navegar do nível de detalhe mais alto para o nível de detalhe mais baiso. Os caminhos das operações de navegação são determinados pelas hierarquias de uma dimensão.

### 3.2.5.2. Slice and dice

*Slice and Dice* são operações para realizar a visualização dos dados em diferentes perspectivas. Slice é a operação que corta o cubo, mas mantém a mesma perspectiva de visualização dos dados. *Dice* é a mudança de perspectiva da visão. Este processo é semelhante a girar o cubo e ver o conteúdo armazenado na outra face.

## 3.3. Considerações finais

Este capítulo apresentou a técnica de modelagem dimensional. Esta técnica tem sido comumente utilizada na modelagem de dados para *data warehouse* em substituição a modelagem E/R muito utilizada em projeto de banco de dados operacionias.

O próximo capítulo discute as ferramentas OLAP apresentando suas características e funcionalidades básicas.

# 4. FERRAMENTAS OLAP

Este capítulo aborda as ferramentas de acesso ao *data warehouse*. Conhecidas como ferramentas OLAP representam, juntamente como o banco de dados, os dois principais componentes do ambiente de *data warehouse*. Através da análise das características e funcionalidades que cada ferramenta oferece pode-se decidir pela melhor alternativa para aquisição da ferramenta desejada. Com o objetivo de tornar o processo de escolha de uma ferramenta OLAP mais simples e compreensiva, este capítulo enumera as diferenças existentes entre o processamento analítico e o processamento transacional.

## 4.1. Definição de OLAP

O processamento analítico *on line* (OLAP) refere-se ao conjunto de processos para criação, gerência e manipulação de dados multidimensionais para análise e visualização pelo usuário em busca de uma maior compreensão destes dados (CAMPOS & ROCHA, 2001). A análise multidimensional é a habilidade de manipular dados que tenham sido agregados em várias categorias ou “dimensões”. De acordo com o Conselho de OLAP, o propósito da análise multidimensional é auxiliar o usuário a sintetizar informações empresariais através da visualização comparativa, personalizada e também por meio da análise de dados históricos e projetados (INMON et al., 1999). É comum o uso da expressão “ferramenta OLAP” referindo-se aos sistemas com estas funcionalidades e que são, juntamente com o SGDB, a base do ambiente de *data warehouse*.

A discussão em torno dos sistemas de *data warehouse* se completa na medida em que incluímos as atividades que este ambiente suporta. Parte da atividade executada sobre o *data warehouse* é predefinida sendo semelhante aos tradicionais sistemas de suporte a decisão. No entanto, processos que exigem uma análise mais complexa e detalhada não estão disponíveis pelos métodos e ferramentas tradicionais. Estes processos mais complexos requerem o uso de ferramentas que implementam essas funcionalidades (HAHN, 2000).

## 4.2. Ferramentas de acesso ao data warehouse

Um dos objetivos do *data warehouse* é torná-lo o mais flexível e aberto quanto possível. Não é desejável que, para o acesso e uso do *data warehouse*, tenha um custo inicial elevado de *software* e treinamento. O *data warehouse* deve ser acessível pelo maior número de usuários e plataformas possível. Contudo não seja possível disponibilizar todas as funcionalidades do *data warehouse* para todas as ferramentas do usuário final. A capacidade de consulta de muitas planilhas de cálculo pode ser adequada para muitos usuários que apenas efetuam consultas simples sobre o *data warehouse*. Outros usuários podem necessitar de uma ferramenta mais poderosa que implemente funcionalidades que permitam efetuar análise multidimensional, por exemplo. Os administradores do *data warehouse* tem a responsabilidade de identificar as ferramentas dentre as diferentes funcionalidades que elas implementam no acesso aos dados. Muitas vezes o caminho progressivo para uma ferramenta de mais alto nível é desejável. Pode-se iniciar usando uma ferramenta familiar aos usuários e a medida que o usuário torna-se familiarizado com o uso do *data warehouse* e dos recursos que ele disponibiliza, pode-se investir para a aquisição de uma ferramenta mais adequada e complexa e que justifique o investimento.

Na maioria dos projetos de *data warehouse*, existe a necessidade de selecionar a ferramenta preferida pelos usuários que mais utilizam o *data warehouse*. Um pequeno número de usuários gera a maior atividade de análise sobre o *data warehouse*. Desta forma, o *data warehouse* pode ser ajustado apropriadamente para contemplar a necessidade destes usuários (GUPTA, 1997).

## 4.3. Processamento OLTP X OLAP

Entender uma ferramenta OLAP implica em distinguir o ambiente operacional em que ela se insere. No processamento transacional temos a predominância de operações de manipulação de dados através dos comandos de inserção, deleção e atualização. Tipicamente, este ambiente é projetado para sofrer o processamento diário dos sistemas.

A Tabela 4.1 mostra as principais diferenças entre os dois tipos de processamento:

| | OLTP | OLAP |
|---|---|---|
| **Tipo de banco de dados** | Relacional | Multidimensional |
| **Dados** | Individualizados | Sumarizados |
| **Tempo** | Presente | Histórico |
| **Quant. De Registros** | Um registro por vez | Muitos registros por vez |
| **Organização dos dados** | Orientados ao processo | Orientados ao Negócio |

TABELA 4.1: Processamento OLTP X OLAP

Quanto ao tipo de banco de dados utilizado temos a predominância do SGBDR (Sistema de Gerenciamento de Banco de Dados Relacional) no ambiente OLTP enquanto que no ambiente OLAP a abordagem multidimensional tem sido mais adequada e está sendo mais aceito no ambiente analítico.

Outra diferença esta na forma como os dados são vistos nesses ambientes. No ambiente OLTP os dados são tratados individualmente, ou seja, não há preocupação quanto ao que esse dado representa como informação para a organização. A única questão relevante para OLTP é o sucesso da transação efetuada com o dado. Já as ferramentas OLAP preocupam-se com a análise dos dados que a empresa possui.

Os sistemas OLTP e *data warehouse* tratam o tempo de forma diferente. O sistema OLTP é um status instantâneo dos negócios de uma organização sendo atualizado constantemente. O *data warehouse* representa uma sequência temporal destes instantâneos.

A orientação dos dados da empresa possui rumos diferentes. As ferramentas OLTP estão ligadas ao nível operacional da empresa através dos processos e procedimentos internos. No outro extremo estão as ferramentas OLAP relacionadas com o nível estratégico da organização. Os dados são transformados em informações requeridas pela administração para a gestão do negócio (RODRIGUES et al., 2001).

## 4.4. Funcionalidades básicas

A escolha de uma ferramenta OLAP envolve a análise das funcionalidades que a ferramenta implementa. Em 1993, E. F. Codd, através de um artigo, publicou um artigo para avaliação de ferramentas OLAP. Nesse artigo, 12 regras foram enumeradas identificando as funcionalidades que os produtos OLAP deveriam conter:

Visão conceitual Multidimensional: A ferramenta deve ser capaz de manipular modelos de dados multidimensionais e não estejam necessariamente armazenados em formato multidimensional.

Transparência: O ambiente OLAP deve ser aberto e transparente para o usuário.

Acessibilidade: A ferramenta OLAP deve criar seu próprio esquema lógico para armazenamento de dados

Performance: Consultas não devem ser degradadas com o aumento das dimensões ou do aumento do banco de dados.

Arquitetura Cliente/servidor: O componente servidor deve suportar vários tipos de clientes garantindo compatibilidade com outros bancos de dados. O servidor deve ser capaz de mapear e consolidar dados de outras fontes de banco de dados através da construção de esquemas conceituais e lógicos e físicos próprios.

Dimensionalidade genérica: As estruturas das dimensões devem equivaler-se. Uma função aplicada a uma dimensão poderá também ser aplicada a outras dimensões.

Manipulação de matrizes esparsas de forma dinâmica: A estrutura física do servidor OLAP deve adaptar-se ao modelo analítico criado. O servidor OLAP deverá ser capaz de deduzir sobre o volume de dados e sua esparsidade, sua distribuição, armazenando-os de forma eficiente.

Multi-usuário: A ferramenta deve permitir concorrência de acesso garantindo integridade e segurança.

Operações entre dimensões: Possibilidade de efetuar operações entre as dimensões.

Manipulação intuitiva de dados: Facilidade na manipulação e análise dos dados apresentados.

Relatórios flexíveis: A partir da solicitação do usuário, a ferramenta dinamicamente manipula os dados e os apresenta ao usuário.

Níveis de dimensões e agregações: A ferramenta deve ser capaz de acomodar de 15 a 20 dimensões.

## 4.5. Classificação das ferramentas OLAP

A classificação das ferramentas OLAP é baseada na alternativa para a sua multidimensionalidade. Desta forma, pode-se dividi-las em ferramentas que utilizam um banco de dados multidimensional (MDDB) e as que armazenam os dados em bancos de dados relacionais (RDDB). Um banco de dados multidimensional oferece um ambiente muito simples e de fácil operacionalização e entendimento para usuários que necessitam da capacidade de analisar “fatias” dos dados em um único local. As estruturas de dados para um banco de dados multidimensional são baseadas nas hierarquias das dimensões e na hierarquia das medidas (MACHADO, 2000).

Na literatura existente (PENDSE, 2001), o termo arquitetura OLAP é utilizado para descrever a forma de armazenamento e recuperação dos dados em sistemas OLAP. Desta forma as arquiteturas são classificadas como ROLAP, MOLAP, HOLAP, DOLAP e WOLAP. Na verdade essas não são arquiteturas e sim formas de armazenamento (GARCIA et al., 2001).

Existe uma grande variedade de fornecedores dessas ferramentas no mercado. Cada qual oferecendo um conjunto de módulos que abrangem parcial ou totalmente as funcionalidades apresentadas no item anterior. A falta de padronização dessas ferramentas é conseqüente da origem das mesmas, podendo ser divididos em grupos: Aqueles que têm como objetivo a criação de uma ferramenta OLAP original e aqueles que desenvolveram suas ferramentas a partir de seus SGBDs existentes. Podem ser considerados como exemplos dentro desses grupos os seguintes:

Fornecedores de SGBDs tradicionais (IBM, ORACLE, MICROSOFT) que agregaram módulos para suportar a multidimensionalidade e ferramentas de extração e limpeza de dados.

Os fornecedores de soluções corporativas tais como ERP, SAP, BAAN. Aos seus ambientes integrados de dados foram adicionados módulos para criação de *data warehouse*s. As ferramentas OLAP neste caso fazem parte do conjunto completo da solução.

Fornecedores de sistemas dedicados ao ambiente OLAP. Estas ferramentas possuem características estruturais tais como modularidade, arquitetura em camadas e interfaces amigáveis.

## 4.6. Estratégias de armazenamento

De acordo com FORSMAN (1997), as tecnologias de *data warehouse* e OLAP são complementares onde a primeira guarda e mantém os dados e a segunda transforma estes dados em informações estratégicas. As ferramentas OLAP podem ser classificadas de acordo com a estratégia de armazenamento em:

ROLAP (Processamento Analítico On Line Relacional): O ROLAP é a utilização da estrutura relacional na solução OLAP. A ferramenta OLAP implementa o modelo dimensional para gerenciar os dados sendo as consultas transformadas em requisições SQL. A vantagem desta ferramenta é a utilização da tecnologia relacional estabelecida, de arquitetura aberta e padronizada.

MOLAP (Processamento Analítico *On Line* Multidimensional): Utilização de banco de dados com características multidimensionais (MDDB). Os dados são armazenados em estruturas que representam os cubos multidimensionais. Os dados são mantidos em estruturas de dados do tipo vetor de maneira a prover um ótimo desempenho no acesso a qualquer dado. A forma de acesso e de agregação dos dados faz com que esta ferramenta tenha um excelente desempenho. Outra vantagem é o rico complexo conjunto de funções de análise que oferece.

HOLAP (Processamento Analítico *On Line* Híbrido): Representa uma abordagem de uso misto das duas estratégias anteriores onde as estruturas relacionais são utilizadas para os dados com maior granularidade em esquemas estrela e as estruturas multidimensionais são utilizadas no armazenamento de menor granularidade, ou seja, nos agregados.

WOLAP (Processamento Analítico *On Line – Web*): As facilidades desta arquitetura residem na possibilidade de usar plataformas independentes para dar suporte a usuários remotos, aplicações de groupware, facilidade de aprendizado e facilidade de manutenção. No entanto, as limitações dos recursos da internet, das interfaces e das funcionalidades, são desvantagens dessa arquitetura .

## 4.7. Considerações finais

Este capítulo abordou as ferramentas de acesso ao *data warehouse*. Apresentamos as características e funcionalidades que cada ferramenta oferece. Com isso o processo de escolha de uma ferramenta OLAP torna-se mais simples e compreensivo. O próximo capítulo apresenta três metodologias extraídas da bibliografia. Após a apresentação das metodologias, ao final do capítulo 5, destaca-se os pontos positivos e negativos de cada uma das metodologias apresentadas.

# 5. METODOLOGIAS PARA PROJETO DE DATA WAREHOUSE

Neste capítulo serão apresentadas três metodologias selecionadas a partir do levantamento bibliográfico. Cabe destacar que o critério adotado para a seleção foi o fato destes trabalhos terem sido elaborados em ambiente acadêmico e que tratam o projeto de *data warehouse* de forma mais abrangente identificando as várias etapas do processo. Neste capítulo, inicialmente, será feita a simples apresentação da metodologia conforme a proposta do respectivo autor. Em seguida será feita uma avaliação das metodologias apresentas destacando seus pontos positivos e negativos, suas contribuições e limitações.

## 5.1. Metodologia segundo MOODY & KORTINK (2000)

O objetivo principal desta metodologia é propor uma alternativa à modelagem dimensional proposta por Kimball. De acordo com KIMBALL (1997) a modelagem de *data warehouse* é completamente diferente da modelagem de sistemas transacionais e que as técnicas de modelagem E/R não podem ser igualmente aplicadas. No entanto, MOODY & KORTINK (2000) argumentam que a modelagem E/R é igualmente aplicável no projeto de *data warehouse* e *data mart*.

Segundo MOODY & KORTINK (2000), a abordagem de projeto proposta por Kimball é introdutória e baseia-se apenas no levatamento de requisitos dos usuários identificando os componentes (fatos relevantes que precisam ser agregados e atributos

dimensionais agregáveis) básicos do esquema estrela. O *data warehouse* resultante não passaria de um conjunto discreto de esquemas estrela. Os autores enumeram os problemas práticos da abordagem de KIMBALL (1997):

- Imprevisibilidade dos requisitos dos usuários resulta em projeto instável;
- Projeto incorreto caso o projetista não entenda os relacionamentos entre os dados;
- Agregação prematura dos dados resulta em perda de informação;
- A abordagem é apresentada através de exemplos ao invés de uma metodologia de projeto concreta.

A proposta de MOODY & KORTINK (2000) para projeto de *data warehouse* baseia-se na derivação de modelos dimensionais a partir de modelos de dados globais amplamente conhecidos como modelos E/R. A metodologia concentra-se em três etapas básicas:

- Classificação das entidades
- Identificação das hierarquias
- Projeto de modelo dimensional

Para ilustrar sua metodologia de projeto o autor inicia com um exemplo de modelo de dados conforme ilustrado na figura 5.1. Esta figura mostra um modelo de dados para uma aplicação de vendas. Tal modelo é muito apropriado, segundo o autor, em sistemas de processamento transacional. O Modelo não contêm redundância e explicitamente mostra os relacionamentos entre os dados. O fato de não conter redundância maximiza as operações de atualização. Na figura 5.1 os atributos realçados indicam as chaves primárias das respectivas entidades.

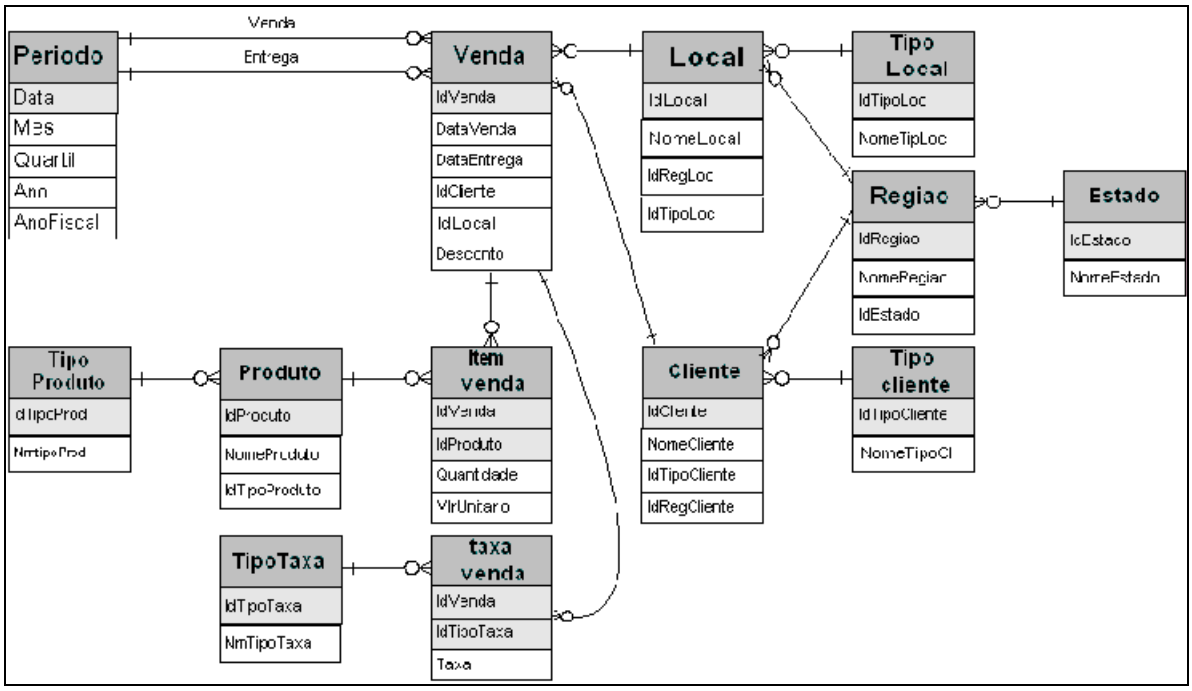



FIGURA 5.1: Modelo de dados exemplo [MOODY & KORTINK,2000]

Infelizmente a maioria dos tomadores de decisão não entende tal esquema. E mesmo consultas simples requerem junções de múltiplas tabelas tornando a consulta complexa. Os usuários então necessitam de pessoal qualificado para escrever as consultas desejadas. A partir desse modelo, MOODY & KORTINK desenvolvem sua metodologia. A seguir vamos descrever as etapas propostas pelo autor.

### 5.1.1. Classificação das entidades

Esta etapa consiste na classificação das entidades. Cada entidade do modelo E/R deve ser incluída em uma das três categorias:

- Entidades de Transação
- Entidades de Componente
- Entidades de Classificação

**Entidades de Transação**: As entidades transação registram os detalhes de eventos relacionados aos negócios da empresa. As principais características destas entidades são: a) descrevem um evento que ocorre em um ponto específico no tempo e b) contém medidas ou quantidades as quais podem ser sumarizadas.

**Entidades Componente:** Uma entidade componente é aquela que está diretamente relacionada com a entidade transação através de um relacionamento "um-para-muitos". As entidades *componente* nos fornecem detalhes de cada transação do negócio (as quais são registradas nas entidades transação). As entidades *componente* respondem as perguntas:

- **Quando** aconteceu o fato ?
- **Quem** é o personagem do fato ?
- **O que** é o objeto do fato ?
- **Onde** aconteceu o fato ?

Os componentes que participam de um evento formam o que chamamos de dimensões do fato. As ocorrências de um fato podem ser analisadas por meio de cada uma de suas dimensões ou ordenadamente através da combinação de dimensões.

**Entidades de Classificação**: As entidades de classificação são as entidades que estão relacionadas com as entidades componentes através de relacionamentos "um-para-muitos", isto é, eles são funcionalmente dependentes das entidades componentes (direta ou transitiva). As entidades de classificação representam hierarquias inseridas no modelo de dados através das entidades componentes para formar tabelas de dimensão em um esquema estrela. A figura 5.2 mostra a classificação das entidades usadas no modelo de dados exemplo. No diagrama, as entidades em **preto** representam **entidades de Transação.** As entidades em **cinza** representam **entidades Componente** e as entidades em **branco** representam **entidades de Classificação.**

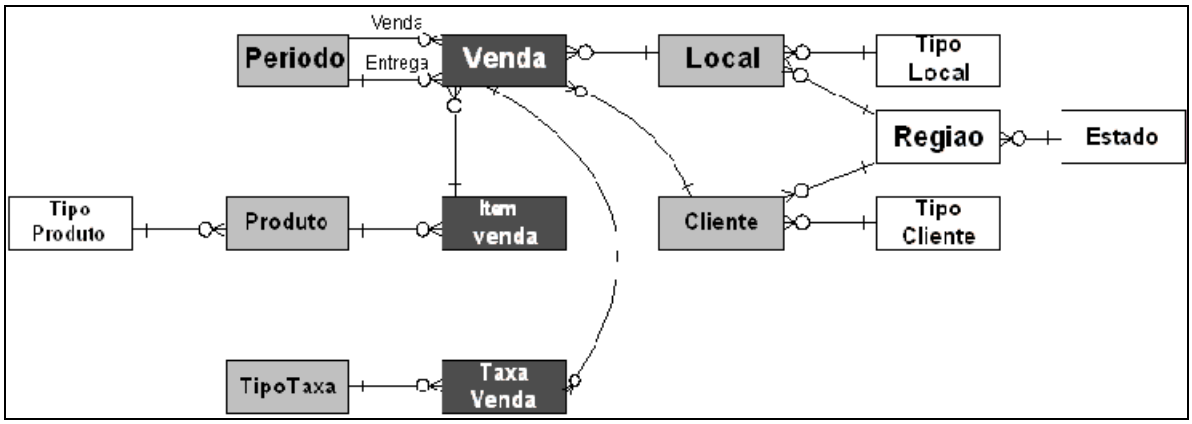



FIGURA 5.2: Classificação das entidades [MOODY & KORTINK,2000]

Em alguns casos, uma entidade pode ser classificada em mais de uma categoria. Neste caso, MOODY & KORTINK (2000) colocam que a ambigüidade é resolvida através da hierarquia de precedência:

1. Entidades de transação (mais alta precedência);
2. Entidades de Classificação;
3. Entidades de Componente (mais baixa precedência).

Ainda, segundo Moody, na prática, algumas entidades não conformam com nenhuma destas categorias. Estas entidades não se ajustam na estrutura do modelo dimensional não podendo ser incluídas em modelos estrela.

### 5.1.2. Identificação das hierarquias

A segunda etapa da metodologia é a identificação das hierarquias existentes no modelo de dados. De acordo com o autor, elas representam a base para a derivação de modelos E/R em modelos dimensionais. Uma hierarquia no modelo E/R é representada por uma seqüência de entidades ligadas através de relacionamentos “um-para-muitos” todos na mesma direção. A figura 5.3 mostra uma hierarquia extraída do modelo de dados exemplo.

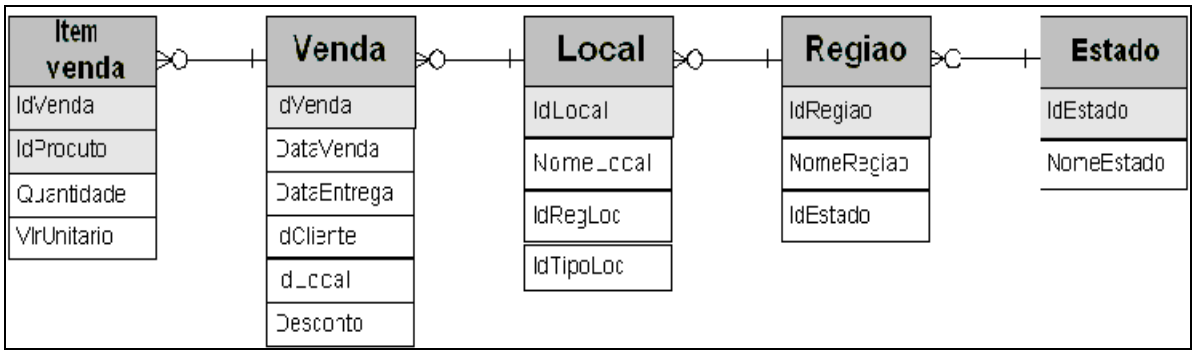



FIGURA 5.3: Exemplo de hierarquia [MOODY & KORTINK,2000]

Na terminologia hierárquica:

A entidade ***Estado*** é "pai" de ***Regiao***

***Regiao*** é "dependente" de ***Estado***

***ItemVenda,Venda, Local*** e ***Regiao,*** são todas descendentes de ***Estado***

***Venda, Local, Regaon*** e ***Estado***, são todas ascendentes de ***ItemVenda***

### 5.1.3. Projeto de modelos dimensionais

Para produzir modelos dimensionais a partir de modelos E/R, MOODY & KORTINK (2000) valem-se de dois operadores: Colapso de hierarquia e Agregação.

**Colapso de hierarquia:** Com este operador, as entidades de nível mais alto podem ser inseridas em entidades de nível mais baixo dentro da hierarquia. A figura 5.4 mostra a entidade *Estado* sendo inserida na entidade *Regiao*. A entidade *Regiao* contém os atributos originas mais os atributos da entidade *Estado*. Esta operação introduz redundância, na forma de dependência transitiva, o qual é uma violação da terceira forma normal (CODD, 1970). A ocultação de um nível na hierarquia das entidades resulta na desnormalização do modelo. A redundância, entretanto, é bastante comum em ambientes de *data warehouse* por resultar em sistemas com desempenho superior comparados aos sistemas que são totalmente normalizados como o esquema floco de neve (*snowflake*).

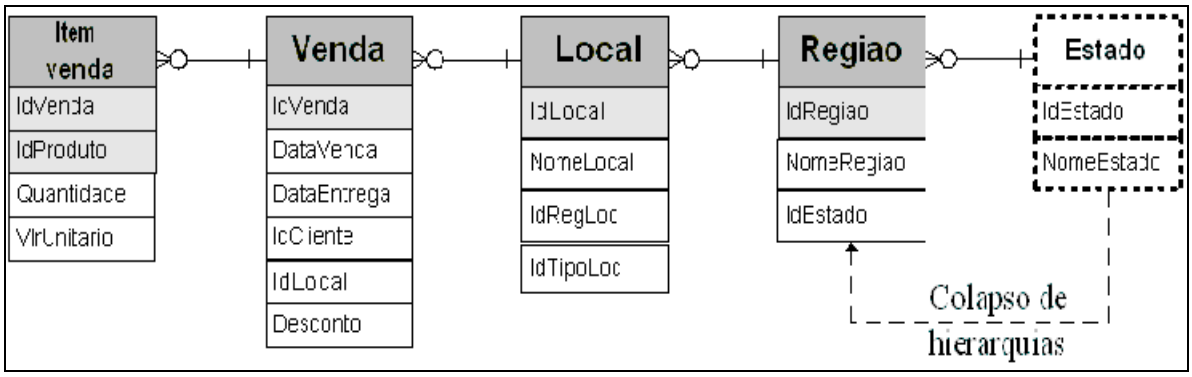



FIGURA 5.4: Operação *colapso de hierarquia* na entidade *State* [MOODY & KORTINK,2000]

**Agregação:** Através do operador de agregação criam-se novas entidades contendo dados sumarizados. Um conjunto de atributos da entidade origem é escolhido para agregar (os atributos de agregação) e um conjunto de atributos destino recebe o agrupamento (atributos de agrupamento). Os atributos de agregação devem ser quantitativos numéricos.

Na figura 5.5 usamos o operador de agregação na entidade *ItemVenda* para criar uma nova entidade chamada *TotaisProduto*. A entidade agregada mostra o *total vendido*, *quantidade média* vendida por pedido e *preço médio* dos items vendidos por produto em uma base diária. Os atributos de agregação são *ValorUnitario* e *quantidade* enquanto que os atributos de agrupamento são *produto* e *data*. A chave desta entidade é a combinação dos atributos usados para o agrupamento. O processo de agregação resulta na perda de detalhes das vendas individuais.

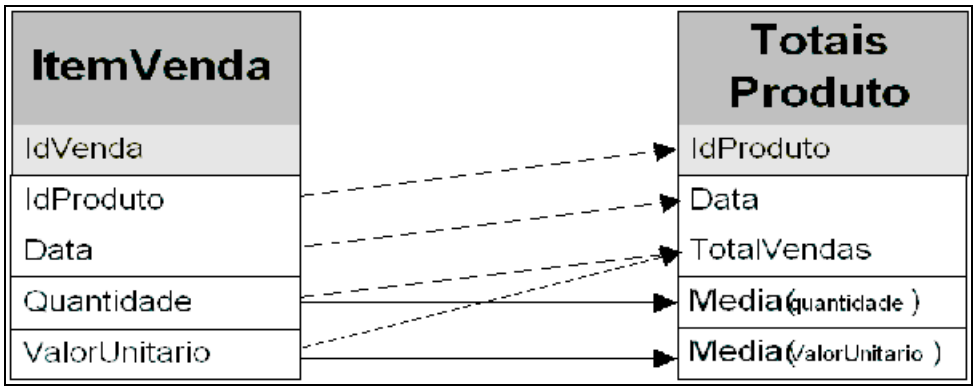



FIGURA 5.5: Operação de agregação [MOODY & KORTINK,2000]

Usando os operadores descritos acima, pode-se produzir diferentes esquemas dimensionais a partir de um modelo E/R. Cada uma das alternativas representa um diferente "*trade-off*" (custo X benefício) entre complexidade e redundância.

A figura 5.6 enumera os diferentes esquemas dimensionais que podem ser utilizados para implementar um *data warehouse*. Quanto mais simples o modelo, maior a redundância. À medida que a redundância é removida, o grau de complexidade do modelo aumenta. Em MOODY & KORTINK (2000), encontra-se a representação de cada um dos esquemas citados.

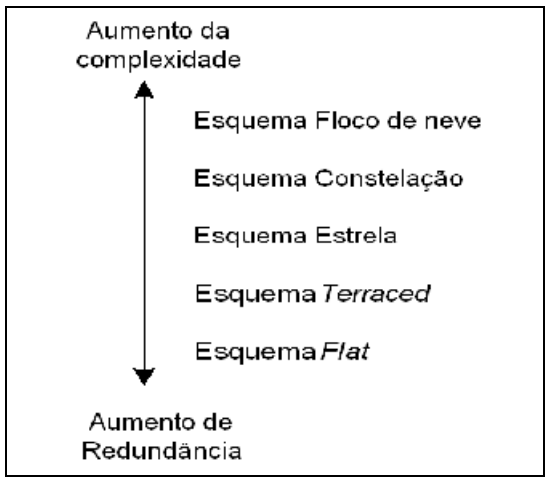



FIGURA 5.6: Opções de esquemas dimensionais para *data warehouse* [MOODY & KORTINK,2000]

MOODY & KORTINK (2000) abordam a criação de esquemas estrela a partir de um modelo E/R da seguinte maneira:

- Uma tabela de fatos é formada para cada entidade transação. A chave da tabela é a combinação das chaves das entidades componente associadas;
- Uma tabela de dimensão é formada para cada entidade componente. A chave desta entidade (tabela de dimensão) será formada pela composição das chaves das entidades de classificação;
- Onde existirem relacionamentos hierárquicos entre as entidades transação, a entidade dependente herda todas as dimensões (e atributos chave) da entidade pai.

Isto provê a habilidade de efetuar operações "*drill-down*" entre os níveis de transação;

- Os atributos numéricos das entidades transação devem ser agregados pelos atributos chave das dimensões.

A figura 5.7 mostra o esquema estrela resultante da entidade fato *Venda*. Este esquema possui quatro dimensões. Cada dimensão possui outras dimensões hierárquicas embutidas através da operação de colapso de hierarquias. O atributo usado para agregação é o total de desconto.

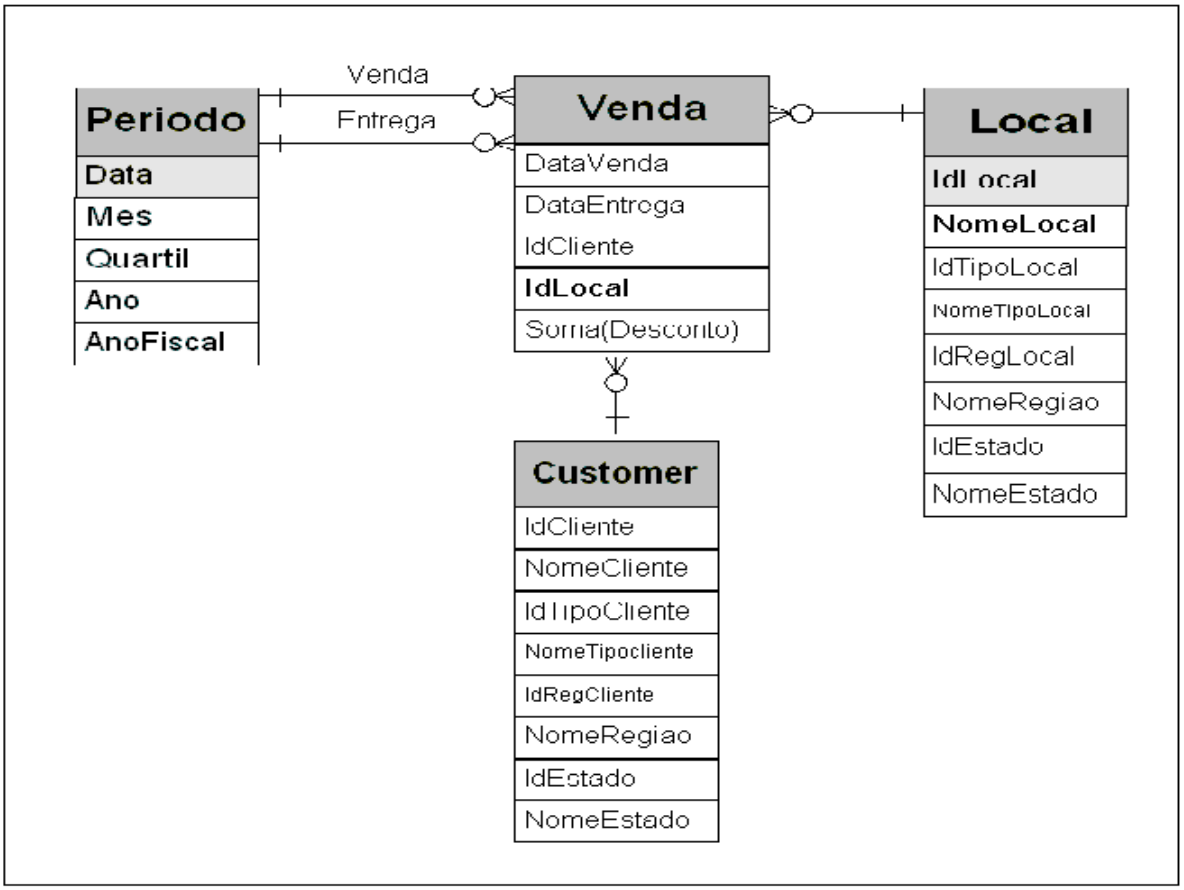



FIGURA 5.7: Esquema estrela do fato "*Venda*" [MOODY & KORTINK,2000]

A figura 5.8 mostra o esquema estrela resultante da entidade *ItemVenda*. Neste esquema temos cinco dimensões. Isto inclui quatro dimensões resultantes da sua entidade

"pai" (*Venda*) e uma dimensão própria (*Produto*). Os fatos agregados foram: "*quantidade*" e "*CustoItem*". A dimensão *Período*, neste modelo, representa duas dimensões. A primeira dimensão compreende a data da venda. A segunda relaciona a data da entrega da mercadoria.

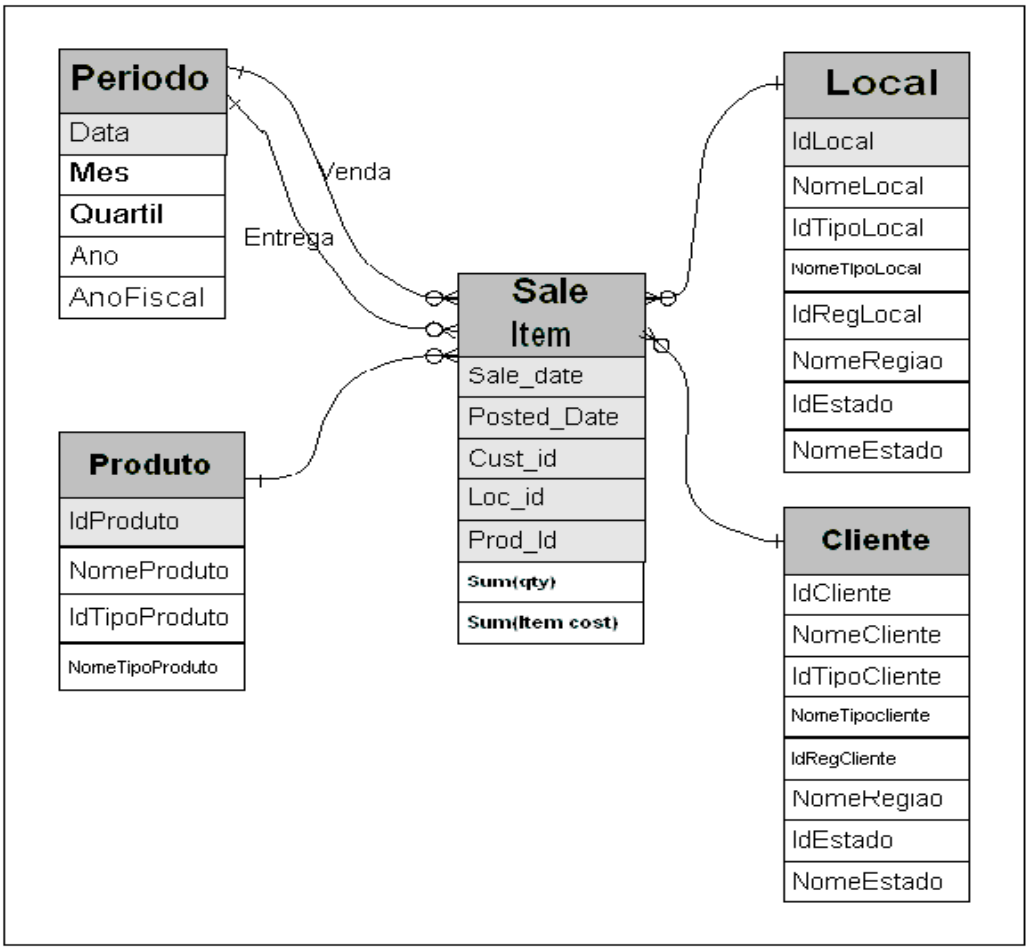



FIGURA 5.8: Esquema estrela do fato "*ItemVenda*" [MOODY & KORTINK,2000]

## 5.1.4. Avaliação e refinamento

A última etapa da metodologia consiste da avaliação e refinamento do modelo dimensional. Na prática, a modelagem dimensional é um processo iterativo. O processo de criação de esquemas dimensionias descrito no tópico 5.1.3, pode ser refinado. As modificações e refinamentos, na sua maioria, estão relacionadas aos padrões não

hierárquicos presentes nos modelos de dados. Os seguintes melhoramentos podem ser aplicados:

Combinar tabelas de fato: As tabelas (de fato) que possuem as mesmas chaves primárias podem ser combinadas em uma única tabela. Isto reduz o número de esquemas estrela, facilitando o trabalho de análise e comparação entre fatos relacionados.

Combinar tabelas de dimensão: A criação de uma tabela de dimensão a partir de cada entidade componente muitas vezes resulta em um grande número de tabelas de dimensão. Para simplificar a estrutura do *data warehouse* as dimensões relacionadas devem ser consolidadas em uma única tabela de dimensão.

Relacionamentos Muitos-para-Muitos: Grande parte da complexidade enfrentada na conversão de modelos E/R para modelos dimensionais resultam dos relacionamentos muitos-para-muitos. Estes relacionamentos causam problemas na modelagem dimensional, pois representam uma “quebra” na estrutura hierárquica e conseqüentemente não podem ser simplificadas diretamente.

## 5.2. Metodologia segundo GOLFARELLI & RIZZI (1998)

Conforme GOLFARELLI & RIZZI (1998), a construção de um *data warehouse* é uma tarefa grande e complexa que requer um planejamento preciso para que possa atender adequadamente as necessidades de informação da organização. Neste sentido, os autores argumentam que a abordagem *bottom up* é mais adequada. O processo inicia através do desenvolvimento do *data mart* mais importante envolvendo a área prioritária e estratégica da organização. Paralelamente, através da disponibilização das primeiras funções do *data mart*, mostra-se aos usuários os potenciais benefícios que este proporciona. Progressivamente, outros *data mart*s são construídos e integrados resultando em um *data warehouse* global.

O autor destaca também a importância de uma ferramenta para auxiliar o projetista nas várias etapas de desenvolvimento. Em GOLFARELLI & RIZZI (2001), o autor desenvolveu um protótipo de uma ferramenta para dar suporte a sua metodologia.

A Tabela 5.1 mostra a seqüência de passos da metodologia proposta pelo autor. A seguir vamos abordar cada uma destas fases.

| ***Passo*** | ***Entrada*** | ***Saída*** | ***Envolve*** |
|---|---|---|---|
| Análise do sistema de informações | Documentação existente | Esquema do banco de dados | Projetista, gerentes dos sistemas de informações |
| Especificação dos requisitos | Esquema de banco de dados | Fatos; carga preliminar | Projetista, usuário final |
| Modelagem conceitual | Esquema de banco de dados, fatos, carga preliminar | Esquema dimensional | Projetista |
| Refinamento e validação do Esquema dimensional | Esquema dimensional; modelo lógico; carga preliminar | Carga | Projetista, usuário final |
| Projeto lógico | Esquema dimensional; modelo lógico | Esquema lógico do DW | Projetista |
| Projeto Físico | Esquema lógico do DW; DMBS destino; carga | Esquema físico do DW | Projetista |

TABELA 5.1: Fases da metodologia segundo GOLFARELLI & RIZZI

### 5.2.1. Análise do sistema de informações

O objetivo desta fase é reunir a documentação dos sistemas de informações existentes na empresa. O projetista com o auxilio das pessoas envolvidas na administração dos sistemas tenta elaborar o esquema de banco de banco de dados (conceitual ou lógico) a partir da documentação coletada.

### 5.2.2. Especificação dos requisitos

A segunda fase da metodologia proposta por GOLFARELLI & RIZZI (1998) consiste em coletar e filtrar os requisitos dos usuários. Esta fase especifica os fatos pertinentes e proporciona uma visão inicial da carga de trabalho envolvida. Os fatos

representam o foco de interesse da empresa e tipicamente correspondem ao registro dos eventos que acontecem na empresa. Um exemplo da ocorrência de um evento é a emissão de uma nota fiscal a qual representa o registro dos itens vendidos na operação de venda.

A documentação E/R dos sistemas existentes, caso exista, é uma fonte para a identificação dos fatos. Ainda, segundo o autor, a carga preliminar é expressa em linguagem pseudo-natural e tem o objetivo de identificar as dimensões e as medidas de interesse da organização.

### 5.2.3. Modelagem conceitual

A modelagem conceitual proposta por GOLFARELLI & RIZZI (1998) consiste de um conjunto de *esquemas fato* cujos componentes principais são: fatos, dimensões e hierarquias. O esquemas fato são construídos a partir da documentação (modelo E/R ou esquema relacional) da estrutura do banco de dados. Para a produzir os esquemas dimensionais o autor propõem o modelo Fato Dimensional DFM (*Dimensional Fact Model*). DFM é uma técnica semi-automática para a criação de esquemas dimensionais conforme veremos a seguir.

### 5.2.4. Apresentação do DFM

Conforme GOLFARELLI et al. (1998), O esquema *fato dimensional* é estruturado na forma de uma árvore cuja raiz é o fato. O fato é representado por um retângulo no qual é registrado o nome do fato e, tipicamente, um ou mais atributos numéricos que mensuram o fato nas diferentes perspectivas. A figura 5.9 exemplifica um esquema para o fato *VENDA* numa cadeia de lojas. *Qtye vendida* e *valor* são os atributos deste fato.

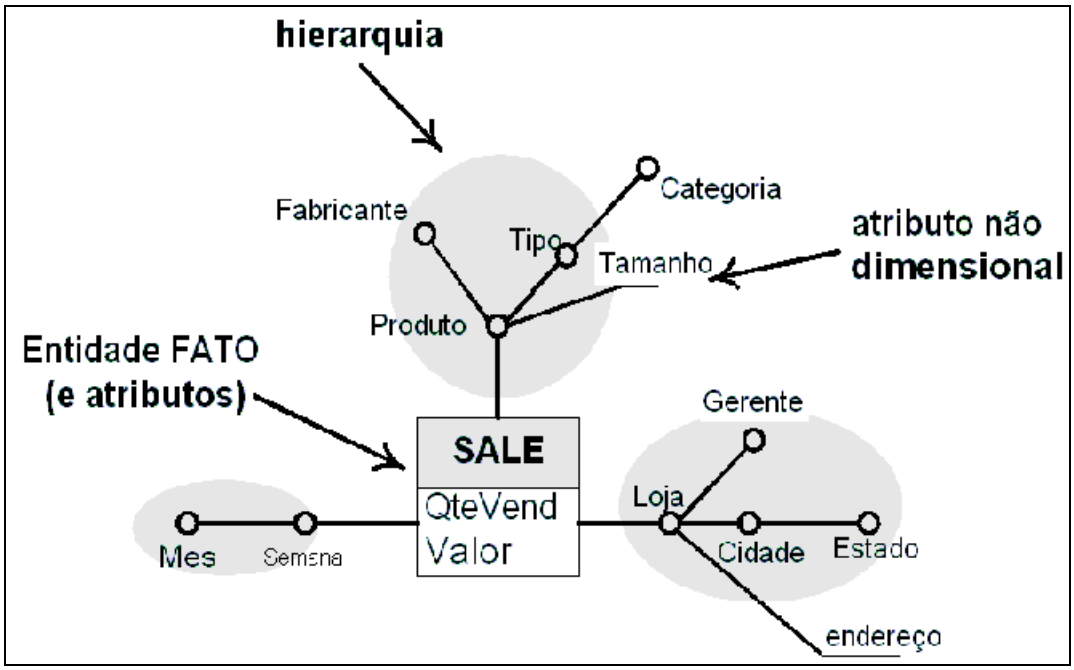



FIGURA 5.9: Um esquema fato tri-dimensional [GOLFARELLI et al.,1998]

Cada vértice ligado diretamente no retângulo (que representa o fato) é uma dimensão. As hierarquias são representadas pelas sub-árvores que fundamentam a dimensão. Os vértices, representados por círculos, são atributos que podem assumir um conjunto de valores discretos. Os arcos representam relacionamentos do tipo “-para um” entre os pares de atributos. A dimensão onde culmina (ligada ao retângulo) a hierarquia defina sua granularidade mínima. Os atributos nos vértices que compõem o caminho da sub-árvore, a partir do fato, representam granularidades gradativamente maiores. O esquema fato da figura 5.9 possui três dimensões: *semana*, *produto* e *loja*.

Alguns vértices terminais podem ser representados através de linhas ao invés de círculos (*tamanho* e *endereço* na figura 5.9). Neste caso eles representam atributos não dimensionais os quais não podem ser usados para agregação. Não faria sentido agregar as vendas de pelo endereço de determinada loja, por exemplo.

Uma instância de um fato (uma linha da tabela de fatos) é representada através de um relacionamento (“muitos-para-muitos”) entre as dimensões. Cada combinação de valores das dimensões define uma instância de um fato. No exemplo, uma instância de um fato descreve a quantidade vendida de um produto em determinada loja em determinada semana. A representação do modelo DFM apresenta as características detalhadas a seguir:

**Agregação**

Através da agregação dos dados no nível de detalhe desejado para as operações de consulta, estas oferecem um desempenho muito superior comparado as consultas nos dados que não estão sumarizados.

A processo de agregação requer a definição de uma função (*sum*() ou *avg*() por exemplo) que será empregada sobre os atributos que caracterizam cada instância gerando um valor que irá representar todo o conjunto. Os atributos dos fatos que serão agregados podem ser aditivos, semi-aditivos ou não aditivos.

**Aditivos**: São os atributos que pode ser somados através da função sum(). Um exemplo de um atributo aditivo no caso do exemplo seria o total vendido para um determinado gerente em todas as lojas que este gerencia.

**Semi-aditivos**: Quando um atributo não pode ser somado ao longo de uma ou mais dimensões. Neste grupo se incluem os atributos que são usados para medir o nível de algo como, por exemplo, a temperatura média do ano ou o nível médio do estoque.

**Não aditivos**: Quando um atributo não é aditivo ao longo de nenhuma dimensão. A temperatura é um exemplo de um atributo não aditivo uma vez que somar duas temperaturas não faria muito sentido. Entretanto, os atributos semi e não aditivos ainda podem ser agregados através das funções mínimo e máximo, por exemplo.

Neste DF (*Dimensinal Fact*) exemplo, os atributos são aditivos ao longo das dimensões (*default*). Atributos semi-aditivos e não-aditivos existentes no modelo serão explicitamente indicados relacionando-os respectivamente com as suas dimensões. Nas situações em que um operador de agregação (diferente de SUM) pode ser usado este será explicitamente indicado no modelo. A figura 5.10 ilustra este esquema.

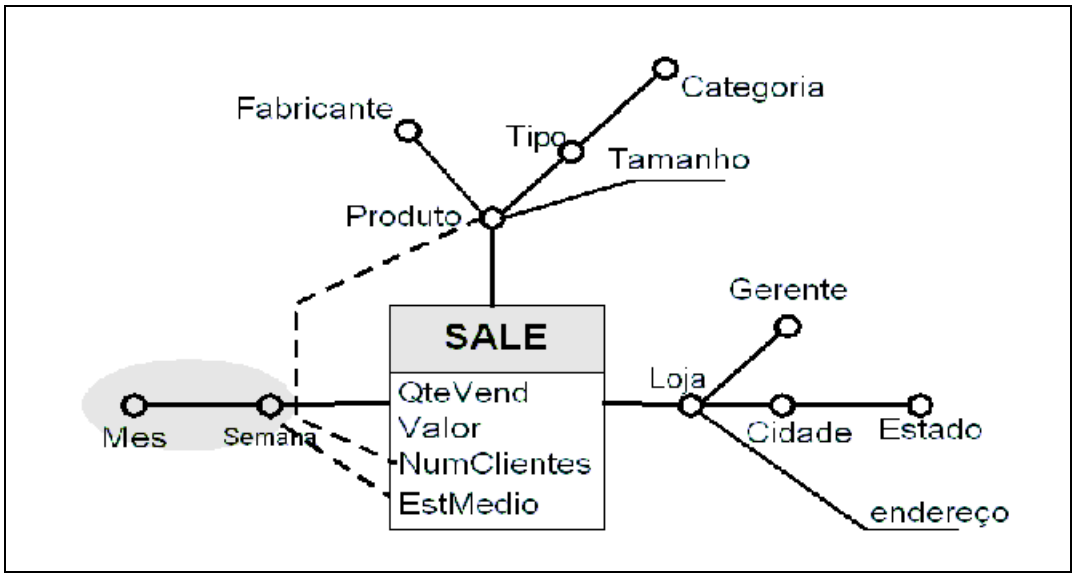



FIGURA 5.10: Um fato semi-aditivo [GOLFARELLI et al.,1998]

**Simplicação de fatos compatíveis**

Conforme GOLFARELLI et al. (1998), dois esquemas de fatos podem ser simplificados (*overlapped*) quando estes forem compatíveis. Isto é, compartilham ao menos um atributo de suas dimensões. Dois esquemas ***F*** e ***G*** podem ser simplificados criando o esquema resultante ***H*** da seguinte maneira:

- O conjunto de atributos de ***H*** é a união de ***F*** e ***G***;
- As dimensões em ***H*** são formadas pela intersecção daquelas em ***F*** e ***G*** contanto que ao menos um atributo da dimensão seja compartilhado;
- Cada hierarquia em ***H*** inclui todas e somente aquelas que são comuns a ambas as hierarquias de ***F*** e ***G***.

Nas figuras 5.11a e 5.11b temos dois esquemas de fatos compatíveis. A primeira representa todos os empregados de uma empresa e a segunda somente os empregados estrangeiros. Estes dois esquemas podem então ser simplificados (Figura 5.11c) formando um novo esquema compatível e simplificado.

No exemplo anterior, mesmo que não esteja explicitamente indicada, a agregação na dimensão tempo pode ter outros atributos. O atributo mês, por exemplo, pode ser convertido para trimestre, quadrimestre, semestre. Estes atributos, conseqüentemente, poderiam ser incluídos na dimensão tempo do esquema resultante. Este procedimento de conversão aumenta o tempo de extração e conversão dos dados. Por este motivo, na fase de projeto, é importante definir adequadamente os atributos das dimensões resultantes.

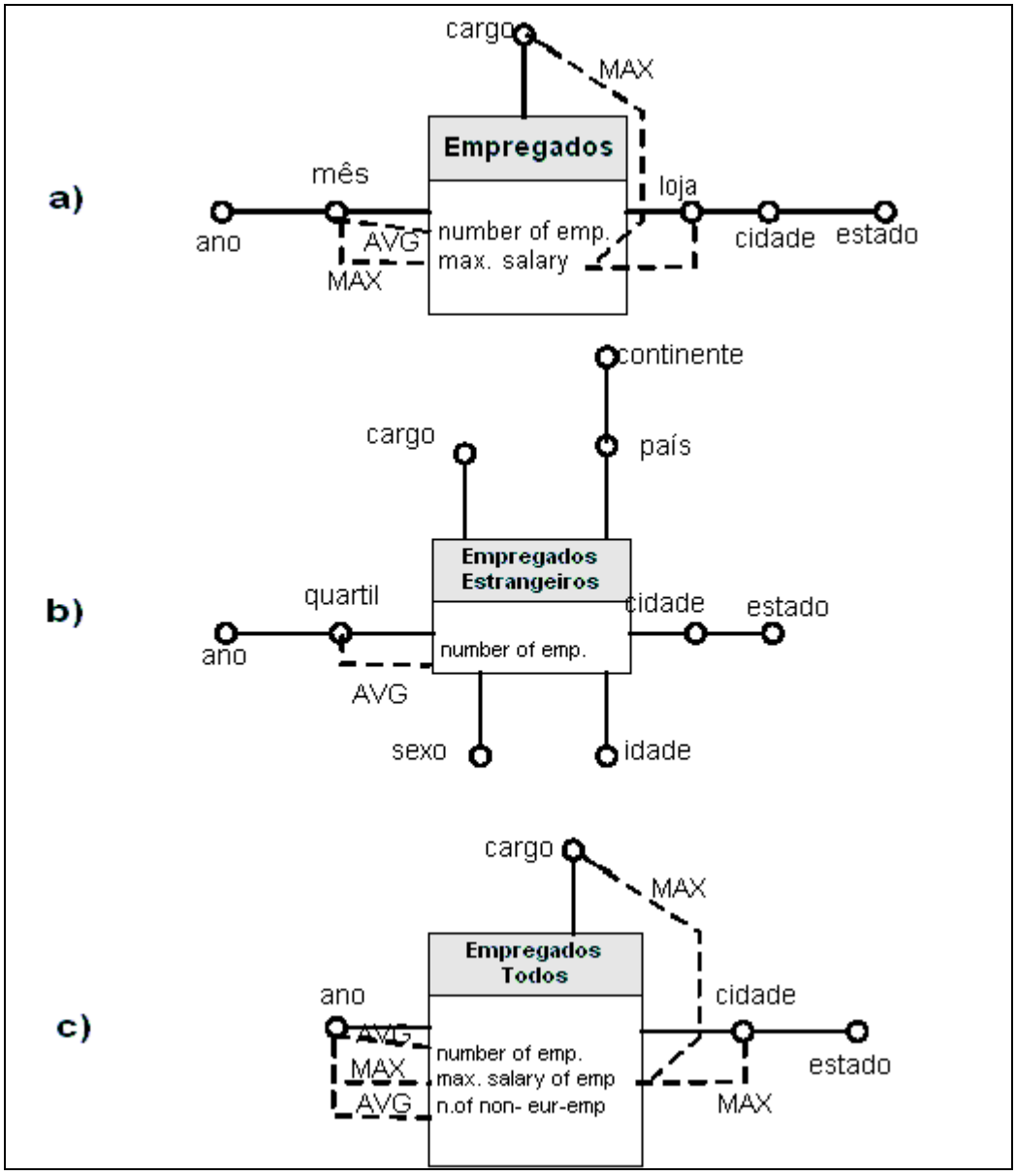



FIGURA 5.11: *Overlap* de esquemas [GOLFARELLI et al.,1998]

## 5.2.5. O processo de criação do DFM

Na seção anterior, foi apresentado modelo DFM (*Dimensional Fact Model*). Golfarelli cita cinco fases para construir tal modelo:

- Construir a árvore de atributos
- Ajustar (cortar e enxertar) a árvore de atributos
- Definir as dimensões
- Definir os atributos dos fatos
- Definir as hierarquias

Para demonstrar o desenvolvimento e aplicação das etapas citadas acima, Golfarelli, utilizar um esquema E/R como exemplo. O esquema mostrado na figura 5.12 representa o fato *VENDA*. Cada instância deste fato corresponde a um produto incluído do cupom de venda. O atributo *PreçoUnitário* foi colocado em *VENDA* ao invés de *PRODUTO*. Isto porque o preço do produto pode variar ao longo do tempo.

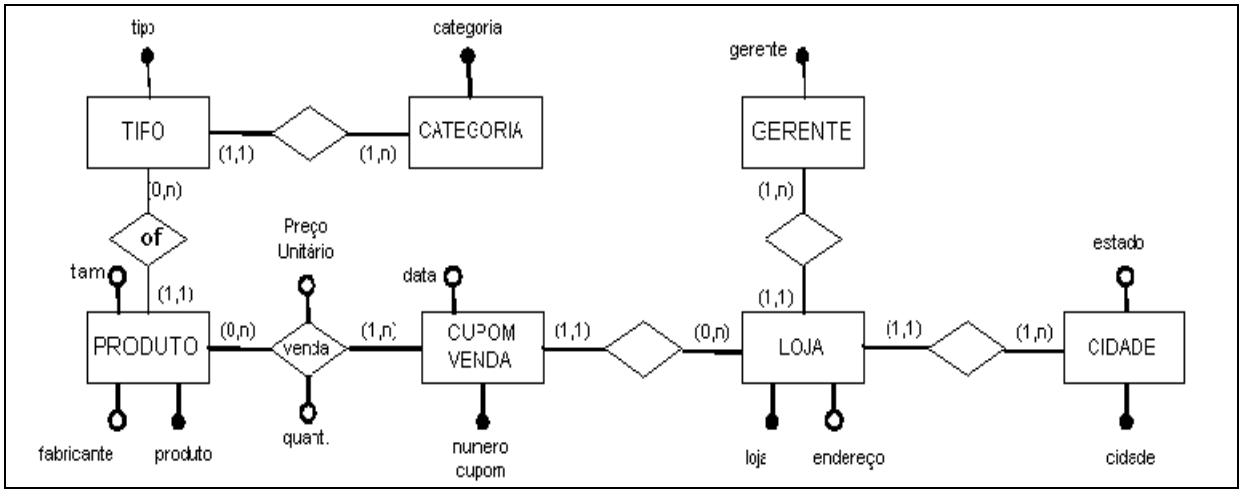



FIGURA 5.12: Esquema E/R para o fato *venda* [GOLFARELLI et al.,1998]

### 5.2.5.1. Definição dos fatos

Os fatos são os elementos principais envolvidos na construção de um *data warehouse*. Eles correspondem ao registro quantitativo e qualitativo de determinado evento hora sendo registrado. Estes eventos ocorrem dinamicamente durante a vida da empresa. No modelo E/R, um fato poder ser representado por uma entidade. Também pode ser representado através de um relacionamento n-para-n. Na prática, este relacionamento n-para-n acaba sendo transformado em uma entidade.

As entidades atualizadas freqüentemente, ou seja, as que sofrem maior atividade de inclusão, são boas candidatas para entidades de fatos. Entidades que sofrem pequeno grau de atualização e que representam propriedades estruturais e domínio dificilmente serão eleitas entidades de fatos (entidade *tipo* e *categoria*)

Através dos esquemas E/R identificamos os fatos. Estes tornam-se a raiz para o novo esquema fato do modelo DF. No exemplo que será desenvolvido a partir de agora, a discussão será concentrada na entidade F representada pelo fato venda. Este fato está representado no esquema E/R pelo relacionamento ***venda*** o qual representa o evento venda de um produto. A figura 5.13 mostra como este relacionamento foi transformado em entidade.

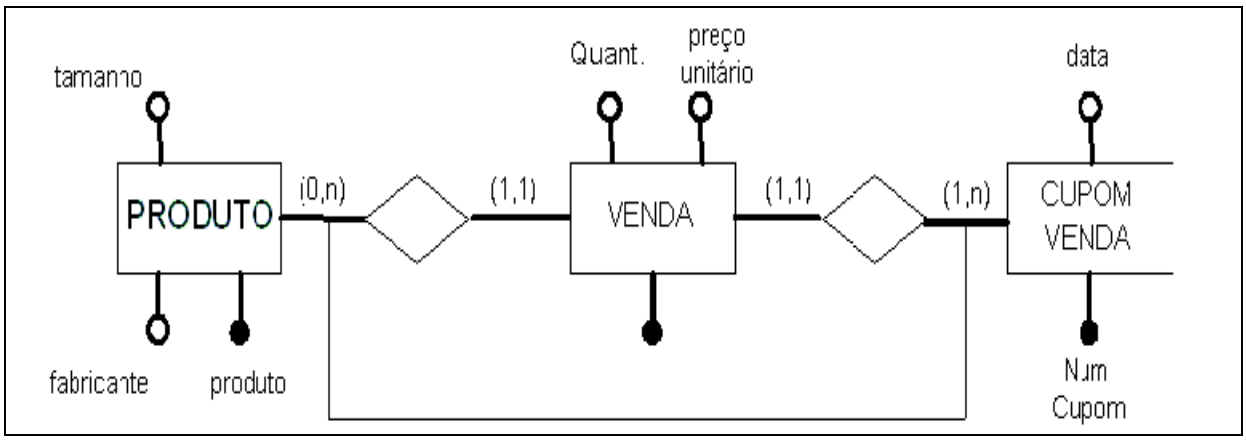



FIGURA 5.13: Transformação do relacionamento *sale* em entidade

### 5.2.5.2. Construir a árvore de atributos

Dada uma parte de um esquema E/R e sendo *F* uma entidade pertencente a esta parte, chamamos árvore de atributos aquela que:

- Cada vértice corresponde a um atributo do esquema;
- A raiz corresponde ao identificador da entidade *F*;
- Para cada vértice ***v***, a funcionalidade correspondente ao atributo determina todos os atributos correspondentes aos descendentes de ***v.***

A árvore de atributos pode ser gerada de forma semi-automática através do uso de um algoritmo. Entretanto, a árvore resultante precisa de alguns ajustes em função de: (GOLFARELLI et al., 1998)

- Existência de relacionamentos muito-para-muitos;
- Transformar relacionamentos um-para-um em atributos não dimensionais;
- Verificação de relacionamentos opcionais do E/R;

Para exemplificar, suponhamos que F seja a entidade escolhida para representar um fato. Através da transformação automática a árvore de atributos resultante correspondente ao esquema E/R é mostrado na figura 5.14.

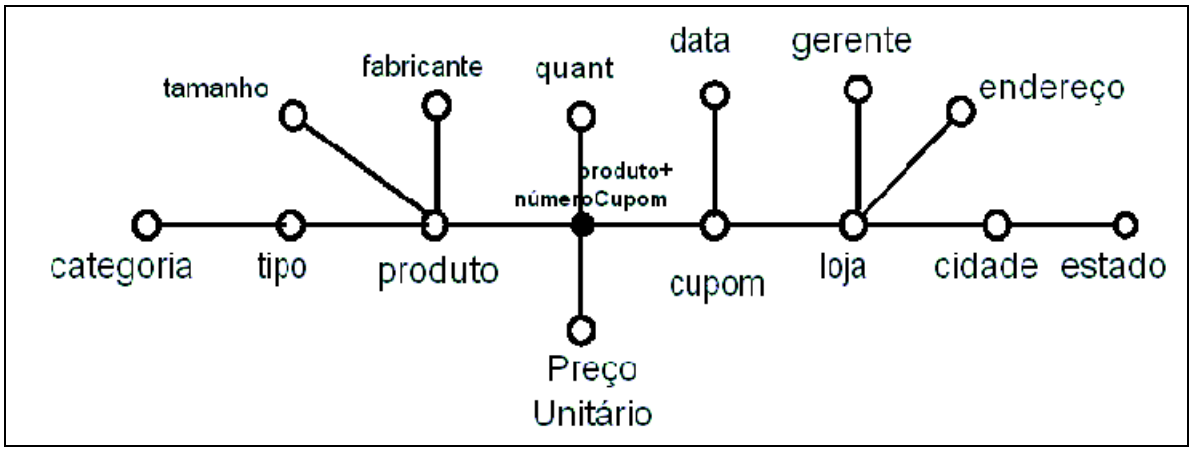



FIGURA 5.14: Árvore de atributos [GOLFARELLI et al.,1998]

### 5.2.5.3. Ajuste da árvore

A etapa de ajuste da árvore de atributos tem a finalidade de eliminar níveis de detalhe desnecessários á um ambiente de *data warehouse*. O processo de ajuste consiste de duas operações:

**Poda**: Tem o objetivo de cortar (podar) as sub-árvores. Os atributos cortados serão eliminados do esquema e conseqüentemente não será possível efetuar agregação de dados sobre estes atributos.

**Enxerto**: O processo de enxerto é usado quando um vértice possui informações que não são importantes, mas seus descendentes precisam ser preservados. Por exemplo, classificar os produtos por categoria não considerando a informação referente ao tipo de produto.

Neste exemplo do fato venda, o *detalhe do cupom* não é uma informação interessante a ser mantida no esquema. Através da operação de poda, a árvore de atributos resulta como mostrado na figura 5.15. Em geral, esta operação resulta na redução da granularidade do *data warehouse*.

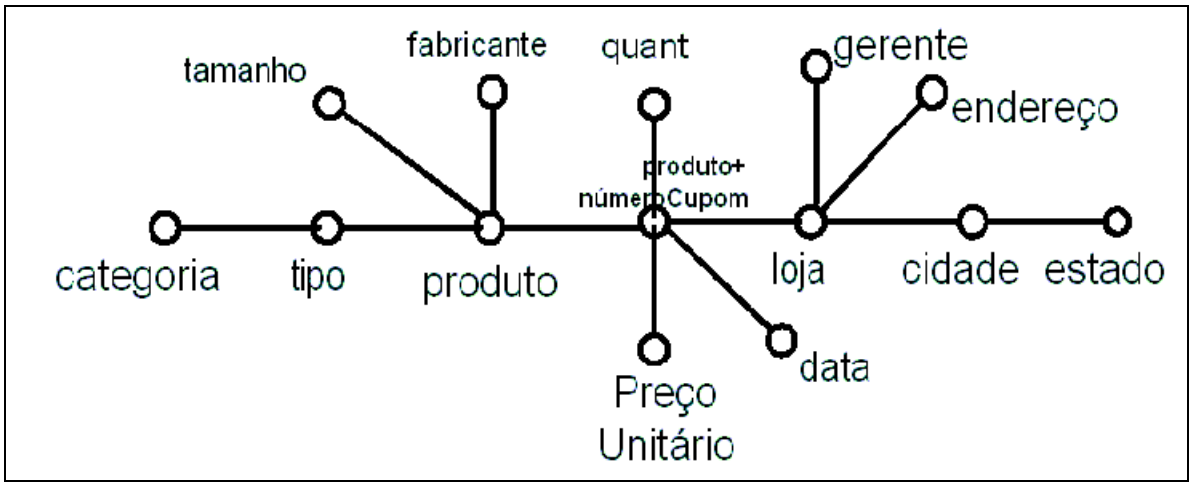



FIGURA 5.15: Árvore de atributos após os ajustes [GOLFARELLI et al.,1998]

### 5.2.5.4. Definição das dimensões

Uma vez criada a árvore de atributos segue-se com a definição das dimensões que farão parte do esquema ora sendo constituído. Os vértices que estão ligados diretamente à raiz da árvore são atributos candidatos a tornar-se dimensões. A escolha dos atributos de dimensão representa uma etapa importante na construção do *data warehouse* pois determina a granularidade das entidades de fatos.

O tempo é uma dimensão reconhecidamente importante no *data warehouse* visto que registra as variações ocorridas num determinado período de tempo bem como permite uma análise da sua evolução. Por este motivo, o atributo *date* foi ligado diretamente à raiz do esquema de fato para representar explicitamente uma dimensão. Além da dimensão tempo, este esquema possui também as dimensões produto e loja como pode ser verificado na figura 5.15. Ao completar esta etapa o esquema fato relativo pode ser desenhado através da adição das dimensões escolhidas a raiz do esquema.

### 5.2.5.5. Definição dos atributos dos fatos

Fatos representam uma medida quantitativa ou qualitativa da ocorrência de determinados eventos. Estes são tipicamente atributos numéricos que acumulam a soma das ocorrências de um fato, ou outra expressão que identifique valores mínimos, médios ou máximos.

Os atributos dos fatos são registrados no esquema (no retângulo que representa o fato). Para facilitar o projeto lógico do esquema fato, é muito aconselhado que seja criado um dicionário de dados (metadados) que explicite o atributo e de que forma ele foi calculado identificando sua origem.

No exemplo aqui sendo desenvolvido temos os seguintes atributos:

*Quantidade vendida=SUM(VENDA.Quantidade)*
*TotalVendido=SUM(VENDA.Quantidade * VENDA.PreçoUnitário)*
*NúmeroDeClientes = COUNT(VENDA)*

### 5.2.5.6. Definição das hierarquias

A definição das hierarquias das dimensões é a última etapa na construção do esquema fato. Ao longo de cada hierarquia, os atributos devem ser arranjados em forma de árvore tal que os relacionamentos x-para-um permaneçam entre cada um dos nós descendentes da hierarquia.

Nesta etapa, a árvore de atributos já demonstra razoável organização em suas hierarquias. Entretanto, ajustes podem ser efetuados com o objetivo de eliminar detalhes irrelevantes do esquema e também, novos níveis de agregação podem ser inseridos para enriquecer o esquema dimensional.

Os atributos que não serão usados para agregação e, portanto, possuem apenas objetivos informativos, podem ser classificados e identificados como atributos não dimensionais.

## 5.2.6. Refinamento e validação do esquema dimensional

Nesta fase é realizado um refinamento da carga preliminar através de sua formulação em maiores detalhes a partir do esquema dimensional produzido. Esta fase também tem o objetivo de validar o esquema dimensional gerado na etapa anterior. A previsão de carga que será gerada no ambiente de *data warehouse* somente terá validade caso o esquema tenha validade comprovada. Se as medidas não foram adequadamente identificadas e as hierarquias bem estruturadas, a previsão de carga não terá sentido. A previsão de carga e volume de dados pode ser computada através da esparsidade.

Conforme GOLFARELLI & RIZZI (1998), um consulta pode ser representada através de um padrão utilizando o modelo fato dimensional. Esta representação consiste de um conjunto de marcas que são colocadas sobre os atributos de dimensão. Estas marcas indicam as agregações que serão efetuadas.

Os dados apresentados pelo resultado de uma consulta podem ser qualquer combinação dos atributos dos fatos ou resultado de expressões de cálculos efetuados na

própria consulta. A figura 5.16 mostra o exemplo que uma consulta padronizada. Ela representa *o total da quantidade vendida e retorno médio por unidade vendida para cada semana e para cada tipo de produto*.

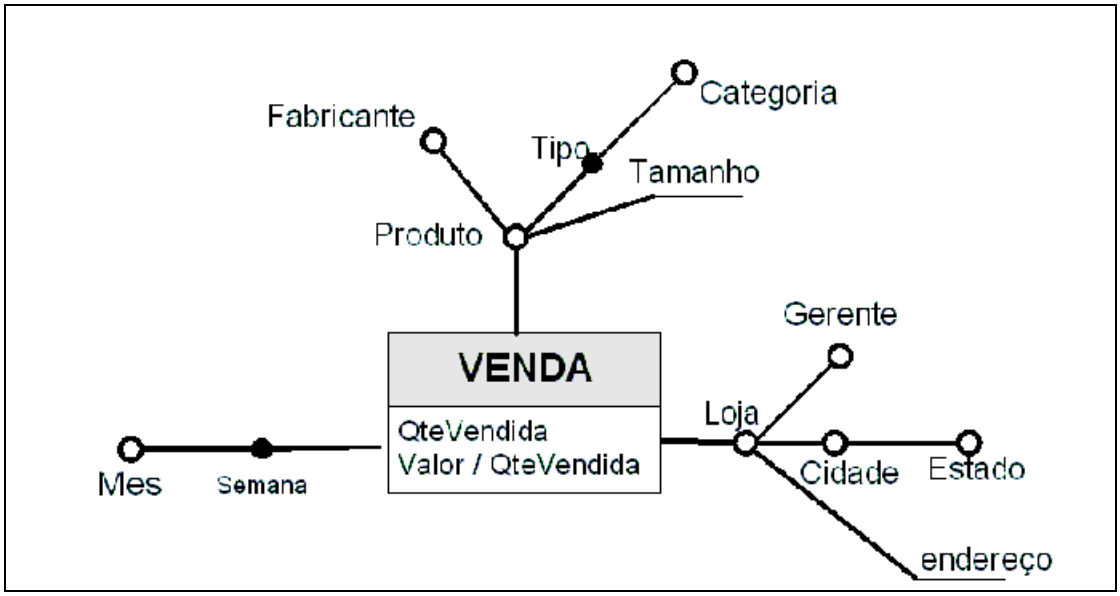



FIGURA 5.16: Representação de um padrão de consulta [GOLFARELLI et al.,1998]

### 5.2.7. Projeto lógico

A fase do projeto lógico recebe como entrada o esquema dimensional ajustado resultado da etapa anterior. Este esquema é convertido para o modelo estrela desenvolvido por (KIMBALL, 1998), considerando sua implementação em uma base de dados relacional. Nesta fase, também são considerados aspectos relacionados à performance do *data warehouse*. (GOLFARELLI & RIZZI, 1998):

- Materialização de visões;
- Transformação do modelo lógico em tabelas de fato e dimensões;
- Particionamento de tabelas de fato.

### 5.2.8. Projeto físico

O aspecto principal a ser considerado na fase de projeto físico relaciona-se a seleção dos índices mais adequados para proporcionar a melhor performance das aplicações do *data warehouse*. Segundo GOLFARELLI & RIZZI (1998), em ambiente de *data warehouse*, a escolha adequada de índices tem complexidade adicional visto que o volume de dados nestes ambientes geralmente é muito maior. Assim, além das tradicionais estruturas de índices B-tree, devem ser consideradas também outras alternativas como, por exemplo: *bitmap index*, *join index* e *projection index.*

## 5.3. Metodologia segundo HERDEM (2000)

A metodologia apresentada em HERDEM (2000) baseia-se na abordagem de modelagem em três níveis: Modelagem conceitual, projeto lógico e projeto físico. Esta abordagem é amplamente difundida e utilizada no desenvolvimento de sistemas transacionais.

Herdem concorda que a abordagem dimensional proposta por Kimball é mais apropriada para o projeto de *data warehouse.* Através da abordagem dimensional, o usário consegue um melhor entendimento do domínio da realidade modelada. E  para construir um *data warehouse* durável, confiável e que atenda aos objetivos e requisistos dos usuários, faz se necessário o uso de uma metodologia de desenvolvimento. Tal metodologia deve valorizar a experiência adquirida com o desenvolvimento de sistemas transacionais considerando os aspectos especiais relativos ao ambiente de *data warehouse*.

### 5.3.1. Ambientes diferentes

Conforme o autor, um banco de dados OLTP e um *data warehouse* possuem diferenças nos seguintes aspectos e que devem ser considerados na metodologia:

- Em base de dados OLTP todos os dados relevantes devem ser modelados enquanto que a modelagem conceitual de *data warehouse* compreende apenas dados necessários para suporte à decisão;
- A modelagem de *data warehouse* contempla o modelo multidimensional;
- O projeto físico de banco de dados OLTP é otimizado para contemplar a performance das aplicações enquanto que banco de dados OLAP deve privilegiar a base para ferramentas OLAP;
- Os metadados têm papel importante no contexto de *data warehousing*;

### 5.3.2. Literatura existente

Com o objetivo de justificar e fundamentar a sua metodologia, o autor classifica a bibliografia existente em quatro grandes grupos:

- Modelagem conceitual multidimensional:
- Transformação para o nível lógico
- Projeto físico do banco de dados
- Metadados

No entanto, segundo o autor, todos os trabalhos citados possuem limitações e deficiências que dificultam ou inviabilizam a sua utilização na prática não sendo considerados como uma metodologia completa uma vez que estes deveriam contemplar todas as etapas do desenvolvimento de um *data warehouse*.

### 5.3.3. Descrição da metodologia

A proposta de metodologia apresentada pelo autor para suprir as deficiências existentes nos trabalhos relacionados consiste das seguintes etapas:

- A definição de um *framework* para o projeto de *data warehouse* onde são apresentadas as principais tarefas no projeto de um *data warehouse*;
- Definição e uma linguagem para a modelagem conceitual. Esta linguagem deve proporcionar a confecção de modelos multidimensionais sofisticados;
- Construção de um algorítmo de transformação do modelo multidimensional para o modelo relacional;
- Projeto físico do *data warehouse*. A aplicação de um método para o projeto físico que considera os aspectos mais importantes desta fase tais como a indexação, particionamento e materialização de *views*;
- Os metadados produzidos pelas etapas anteriores devem ser incluídos em um repositório de metadados;
- Um protótipo para implementar a metodologia apresentada. A metodologia deve dar suporte ao projeto durante as várias fases do desenvolvimento. Para tanto, a metodologia pretende estender a funcionalidade de produtos existentes no mercado.
- Avaliação dos resultados. A última fase pretende efetuar uma avaliação dos resultados obtidos pela aplicação prática da metodologia.

A figura 5.17 esboça a metodologia de projeto proposta por Herdem. A primeira camada trata do modelo conceitual.

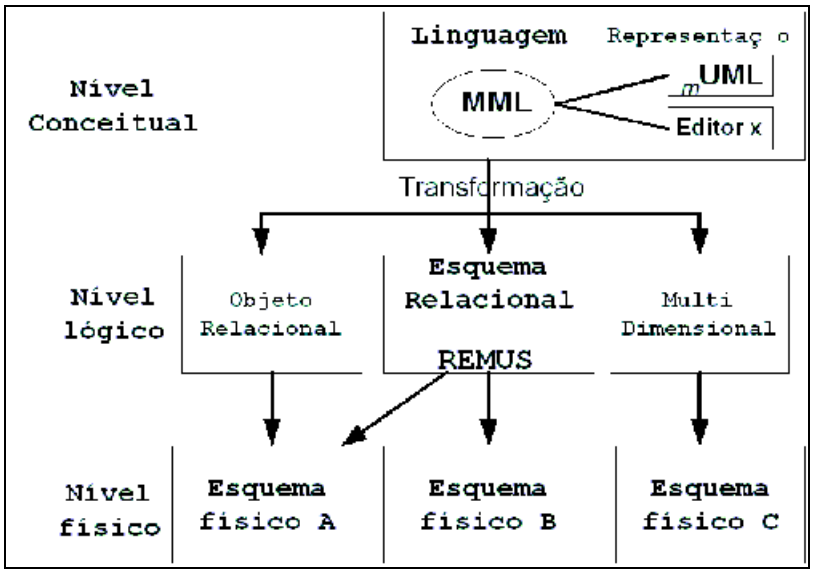



FIGURA 5.17: Modelagem em três níveis [HERDEM, 2000]

Para apoiar esta fase de desenvolvimento (modelagem conceitual) é utilizada a linguagem MML (*Multidimensional Modeling Language*).

As principais características desta linguagem são:

- Orientação a objeto. Por esta razão torna-a uma modelagem flexível e de implementação independente;
- Permite modelar sofisticados modelos multidimensionais pelo fato de distinguir entre classes (fatos e dimensões);
- Permite a evolução dos esquemas através da inclusão de intervalos de tempo aos elementos de conexão.

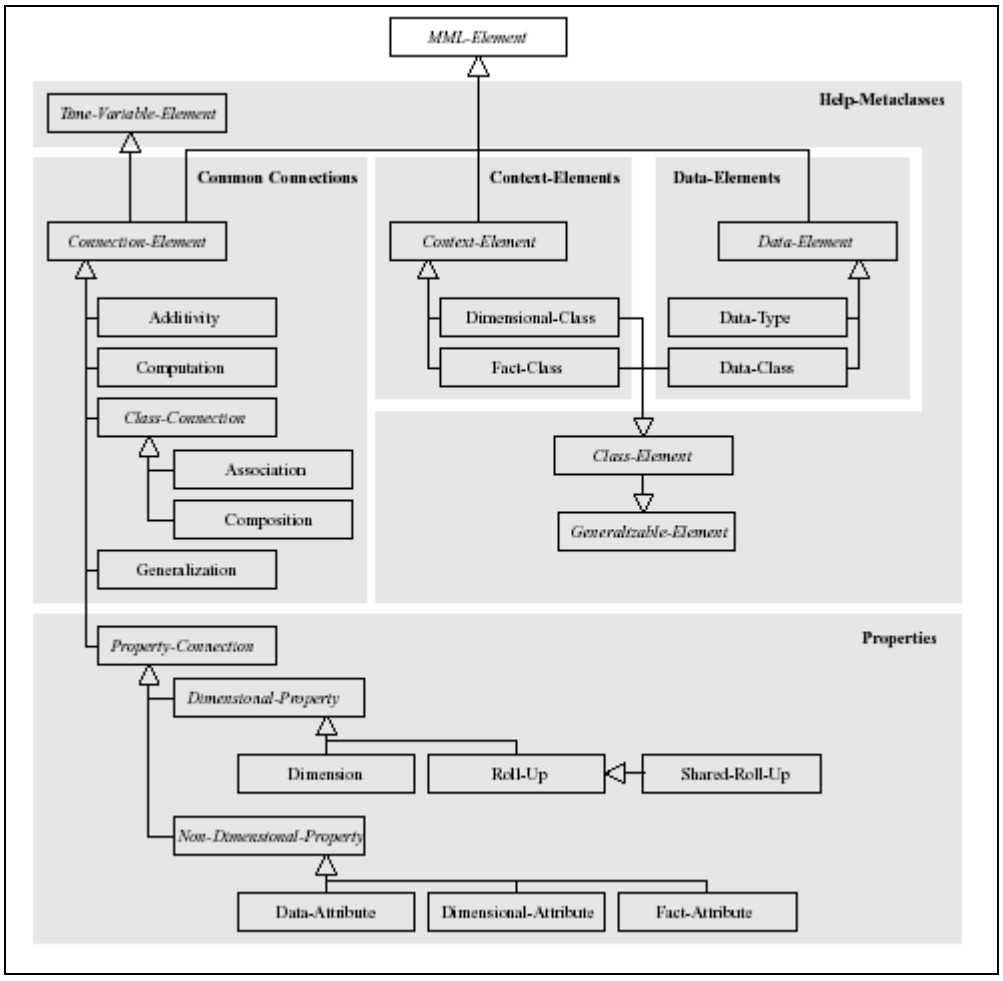



FIGURA 5.18: Hierarquia de herança da linguagem MML [HERDEM, 2000]

A linguagem MML é especificada de forma semiformal através de um diagrama UML (*Unified Modeling Language*) no qual é definida toda a hierarquia de herança seguindo a filosofia de herança das linguagens OO (Orientação a Objeto). A Figura 5.18, extraída de HERDEM (2000) mostra esta hierarquia e os elementos que compõem o esquema. Observa-se que usando MML como base, outras ferramentas para representar o modelo podem ser usadas uma vez que há uma clara separação entre a linguagem de modelagem utilizada e a sua representação gráfica. Especificamente, nesta metodologia, Herdem usa uma extensão de UML (*Unified Modeling Language*) chamada $_m$UML (*Multidimensional UML*) para realizar esta representação.

Os novos tipos de classes (fatos e dimensões) e as suas conexões (*roll-up*) são implementados através do conceito de estereótipos que possibilitam estender a UML.

A arquitetura para implementação desta metodologia é mostrada na Figura 5.19 a qual foi extraída de HERDEM (2000). Observa-se nesta figura que a ferramenta *case Rational Rose* foi estendida através da incorporação de $_m$UML.

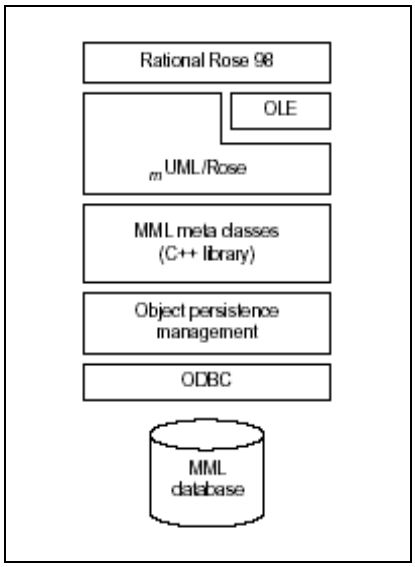



FIGURA 5.19: Arquitetura de implementação [HERDEM, 2000]

O mapeamento do esquema conceitual (feito em MML) para o modelo lógico (banco de dados relacional) é realizado pela a transformação dos diagramas MML para o esquema REMUS (*Relational Model for Multidimensional Purpose*). O esquema REMUS consiste de relações, atributos e metadados. Os metadados carregam as informações das características multidimensionais as quais não podem ser mapeadas diretamente para tabelas e atributos (HERDEM, 2000).

Segundo o autor, o trabalho relacionada a implementação física está apenas iniciando e consistirá das seguintes etapas:

- Um algoritmo transformará o projeto lógico (REMUS) em um esquema físico contemplando a independência de arquitetura de banco de dados a ser usada e

também a independência com relação às ferramentas OLAP a serem usadas pelo o usuário final;

- Um algoritmo que otimiza o esquema resultante considerando aspectos especiais como desnormalização comum em esquemas multidimensionais. Aspectos especiais de refinamento e ajuste devem ser feitos manualmente;
- A última etapa irá contemplar aspectos físicos do banco de dados tais como números de linhas de uma tabela, organização dos discos, e padrões de consulta das aplicações com o objetivo de otimizar o acesso aos dados.

A apresentação da metodologia termina com a preocupação em estender o modelo OIM (*Microsoft Open Information Model*) para os metadados para armazenar detalhes da implementação física do projeto.

## 5.4. Análise das metodologias apresentadas

Após a apresentação das metodologias propostas pelos respectivos autores, segue abaixo uma análise destacando os pontos positivos e negativos considerando a sua aplicabilidade e usabilidade prática em projetos de *data warehouse*.

### 5.4.1. A metodologia de MOODY & KORTINK (2000)

Esta metodologia de desenvolvimento aé mais apropriada para a construção de *data mart*s. A metodologia baseia-se na existência prévia de um *data warehouse* centralizado que reflete o modelo de dados global da organização. Esta abordagem de desenvolvimento de *data mart*s pressupõem também que o *data warehouse* (fonte para o *data mart*) resida em uma base de dados relacional onde as estruturas de dados estejam normalizadas. A metodologia apresenta uma boa abordagem para derivar esquemas dimensionais a partir do modelo de dados existente na empresa. No entanto, para uma organização que não possui um modelo de dados global ou que possua fontes de dados armazenadas em bases não

relacionais, existe a necessidade de realizar primeiramente o projeto de *data warehouse*. Este *data warehouse* servirá para integrar os dados oriundos das várias fontes de forma modular. Posteriormente, a metodologia pode ser aplicada para a construção de *data mart*s.

Um aspecto importante relacionado à fase de levantamento de requisitos dos usuários não é considerado nesta metodologia. Isto porquê, o projeto esta sendo guiado com ênfase nos dados existentes e não através do levantamento de informações sobre os reais requisitos de informações dos usuários.

Outros aspectos fundamentais no projeto de um ambiente de *data warehouse* tais como metadados, granularidade e projeto físico não são abordados nesta metodologia.

### 5.4.2. A metodologia de GOLFARELLI & RIZZI (1998)

A metodologia apresentada por GOLFARELLI (1998) apresenta seis fases distintas. Entretanto não há objetividade nas quatro fases iniciais da metodologia. Estas quatro fases poderiam ser unificadas e tratadas de forma única uma vez que elas representam a fase da modelagem conceitual e antecedem os projetos lógico e físico do *data warehouse*. Além disso, estas quatro fases iniciais são pouco detalhadas pelo autor. A novidade da metodologia de GOLFARELLI (1998) está no desenvolvimento do modelo DMF o qual apresenta bom detalhamento embora a sua compreensão não seja tão simples.

O modelo DFM criado pelo autor possui uma boa fundamentação teórica. No entanto sua aplicabilidade prática é inviável pela ausência de ferramentas de desenvolvimento de *data warehouse* que suportam este modelo.

As fases de projeto lógico e projeto físico são apenas tratadas de forma superficial. A fase de projeto lógico consiste da aplicação da modelagem dimensional proposta por KIMBALL (1996).

Um ponto importante destacado pelo autor é a abrangência e complexidade dos projetos de *data warehouse*. A complexidade e o risco de falha dos projetos podem ser diminuídos através da abordagem de desenvolvimento *bottom up*. A partir da experiência adquirida na construção dos primeiros *data mart*s e da percepção dos benefícios do

ambiente de *data warehouse*, outros *data mart*s são desenvolvidos. Assim, a expansão natural deste ambiente acaba tornando-se uma solução global.

### 5.4.3. A metodologia de HERDEN (2000)

O objetivo do trabalho de HERDEM (2000) é o desenvolvimento de uma metodologia para projeto de *data warehouse* baseado na experiência adquirida ao longo dos anos no projeto de sistemas OLTP. O ciclo de vida de um sistema transacional pode classificado com sendo em três fases. O modelo conceitual, o projeto lógico e finalmente o projeto físico. Para o projeto de sistemas de *data warehouse*, segundo o autor, essas três fases também podem ser seguidas. Os metadados são criados ao longo do projeto de forma incremental. A principal preocupação do autor na elaboração da sua metodologia está na necessidade de uma ferramenta que guie o projetista ao longo do processo. Neste sentido, o autor estende a funcionalidade de uma ferramenta *CASE* (UML) disponível no mercado para possibilitar a modelagem multidimensional.

No entanto, na fase de modelagem conceitual, o autor usa a linguagem MML. Esta linguagem possui características de orientação a objeto. Isso traz um elemento adicional ao já complexo processo de *data warehousing*. Ou seja, o projetista na fase de elaboração do modelo conceitual necessita de conhecimentos de orientação a objeto para criar a especificação do seu modelo. Isto sem dúvida é um fator complicador uma vez que este modelo é implementado em uma base relacional. Neste caso seria prudente a utilização de uma ferramenta que possibilitasse o uso da modelagem E/R.

A metodologia possui dois aspectos positivos. O primeiro aspecto é uma boa visão geral que pode servir de guia para o projetista. A partir da metodologia proposta, o autor identifica as principais ações a serem tomadas para a construção do *data warehouse*. O segundo ponto é a clara distinção entre as fases do projeto. Isso diminui a complexidade do projeto global. A partir das fases enumeradas pelo autor, o projetista deve procurar auxilio na bibliografia procurando maior detalhamento em cada uma das fases do processo.

Entretanto, o ponto negativo da metodologia e a ausência de detalhes referente a cada uma das etapas do projeto. O autor limita a sua metodologia na apresentação de um

esquema genérico (*framework*) apresentando os tópicos gerais que envolvem a construção de um *data warehouse*.

## 5.5. Avaliação geral

Independente da quantidade de aplicações de *data warehousing* atualmente sendo construídas, como disciplina e metodologia, ela ainda precisa e continua a evoluir (MOORE & WELLS, 2000). Isto se deve principalmente ao fato de que o projeto de um *data warehouse* ser complexo envolvendo uma grande variedade de processos, técnicas de desenvolvimento, ferramentas em contínua evolução e pessoal altamente qualificado e conhecedor do negócio da empresa e das suas necessidades de informação.

Observa-se, entretanto, pela quantidade de trabalhos que estão sendo apresentados em congressos, que a pesquisa no campo de *data warehouse* continua evoluindo e despertando grande interesse pela comunidade acadêmica. Outro indicador desta evolução é o volume de investimentos que a indústria (*software* e *hardware*) tem feito nesta área (VASSILIADIS, 2000). Há uma variedade de ferramentas disponíveis para auxiliar o projetista na resolução de problemas específicos. Entretanto, a combinação destas soluções parciais e muitas vezes muito formais e abstratas em uma metodologia mais abrangente ainda está em aberto (GATZIUI et al., 1999).

Muitas obras foram pesquisadas com o objetivo de ampliar as possibilidades e encontrar uma metodologia consistente e de provada aplicabilidade prática. Existem metodologias consagradas e amplamente sendo utilizadas no desenvolvimento de sistemas transacionais, mas, uma metodologia completa e consistente para o desenvolvimento de sistemas de *data warehouse* ainda está para ser elaborada (SCHOUTEN,1999).

Além das três metodologias apresentadas, CAMPOS e BORGES (2002) abordam a construção incremental de *data warehouses* a partir de *data marts*. Em POE et al. (1998) encontramos um bom referencial teórico abordando todo o ciclo de vida de um *data warehouse*. A obra apresenta uma boa base teórica sobre o assunto. Através de experiências práticas obtidas pelo autor, várias dicas são encontradas na obra e que podem ser de grande auxílio ao projetista de *data warehouse*. Também em KIMBALL (1998) encontramos uma

rica apresentação do modelo dimensional com vários exemplos de modelagem multidimensional. Ambas as obras são contribuições importantes e apresentam subsídios e discussões que auxiliam no processo de data warehousing. No entanto carecem de uma metodologia mais abrangente. As obras citadas cobrem apenas tópicos específicos e pontuais e não podem ser consideradas como uma metodologia propriamente dita.

## 5.6. Considerações finais

Conforme apresentado as três metodologias acima descritas, não são suficientes para serem adotadas como guia de implementação. Entretanto, a metodologia apresentada por HERDEM (2000) apresenta-se como a melhor alternativa e pode ser adaptada e complementada a partir de trabalhos de outros autores. A metodologia proposta por GOLFARELLI & RIZZI (2000), contribui significativamente com a proposta de implementação gradual (*bottom up*) do *data warehouse*. A vantagem da implementação gradual é o baixo custo inicial e a possibilidade de aprendizado da equipe de desenvolvimento, reduzindo assim, as possibilidades de insucesso do projeto.

No próximo capítulo apresentamos uma proposta de metodologia que baseiada nos pontos positivos das metodologias apreentadas. A proposta é complementada a partir de trabalhos de outros autores disponíveis na literatura e que representam contribuições importantes no processo de construção de um *data warehouse.*

# 6. PROPOSTA DE METODOLOGIA PARA DESENVOLVIMENTO DE DATA WAREHOUSE

O objetivo deste capítulo é elaborar uma metodologia de desenvolviemnto de *data warehouse*. Na metodologia proposta, dois aspectos têm importância fundamental. O primeiro aspecto é a identificação clara das várias fases envolvidas no projeto de *data warehouse*. O segundo é oferecer um grau maior de detalhamento para que possa ser utilizado como guia na implementação de sistemas de *data warehouse*. Pretende-se com este trabalho, fornecer aos profissionais envolvidos na tarefa de construção de sistemas de data warehouse, uma referência metodológica para auxiliá-los no trabalho de implementação de sistemas de data warehouse

## 6.1. Fases da metodologia

A metodologia descrita em HERDEM (2000) possui uma boa fundamentação e possui três fases distintas: Modelagem conceitual, projeto lógico e projeto físico. A proposta que será desenvolvida aqui acrescenta a metodologia um componente de gerência de projeto e a fase de definição de projeto ambos consideramos importantes para o sucesso de qualquer projeto.

Desta forma a metodologia aqui proposta compreendem as seguintes etapas:

- Gerência do projeto

- Definição do projeto
- Modelagem conceitual
- Projeto lógico
- Projeto físico

Cabe destacar que estas etapas são muito familiares e são utilizadas para o desenvolvimento de sistemas tradicionais. Embora o desenvolvimento de um *data warehouse* possua aspectos diferenciados com relação aos sistemas tradicionais muitas das lições aprendidas no desenvolvimento de sistemas OLTP são de grande valia e devem ser utilizadas no projeto de um sistema de *data warehouse*.

A figura 6.1 mostra as fases da metodologia que compõem o ciclo de desenvolvimento de *data warehouse.*

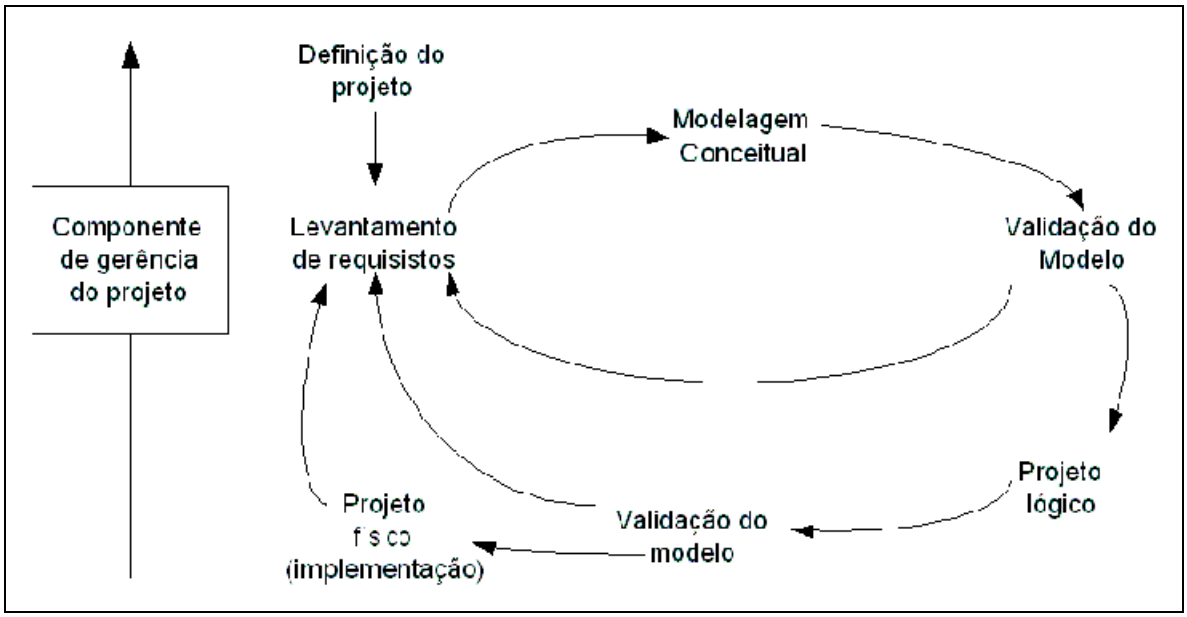



FIGURA 6.1: Fases da metodologia de desenvolvimento de *data warehouse*

O principal aspecto a ser considerado na figura acima é a natureza iterativa do desenvolvimento do *data warehouse*. Isto, mais do que qualquer outra coisa distingue o ciclo de vida de um projeto de *data warehouse* de outros projetos de desenvolvimento. É

sabido que todos os projetos possuem determinado grau de iteração, mas, o projetos de *data warehouse* tem ênfase extrema neste aspecto com o objetivo de rapidamente liberar partes do *data warehouse* para o usuário. Ou seja, enquanto uma parte do *data warehouse* esteja sendo usado pelos usuários outra parte pode estar sendo desenvolvida. Na maioria dos casos prover o usuário com algumas funções do *data warehouse* tem benefícios imediatos (BALLARD et al., 1998).

### 6.1.1. Gerência do projeto

O projeto de *data warehouse*, assim como qualquer outro projeto, também possui um componente responsável pela gerência do projeto. Entretanto, a gerência corresponde ao projeto e não ao *data warehouse*. A diferença fundamental reside no fato de que a gerência do projeto possui um escopo finito enquanto que a gerência do *data warehouse* é similar a gerência de outros processos da instituição (vestibular, matrículas) isto é, preocupa-se com a gerência dos processos de *data warehousing*.

O componente de gerência do projeto têm a responsabilidade de estabelecer o plano geral do projeto. Este plano deve ser conhecido por todos os membros que farão parte da equipe de desenvolvimento do projeto. O plano deve estabelecer o prazo do projeto, os recursos disponíveis e principalmente a expectativa dos usuários com relação ao projeto ora sendo iniciado.

O maior desafio desta etapa é a identificação do responsável pela gerência do projeto. O gerente de projeto tem a responsabilidade de estabelecer as pricipais variáveis do projeto incluindo:

- As funções que o *data warehouse* irá disponibilizar;
- Alocação de recursos (máquinas, ferramentas, pessoas);
- Qualidade (definição de prazos não realísticos podem levar a equipe a seguir atalhos e comprometer a qualidade do *data warehouse).*

### 6.1.2. Definição e denominação do projeto

Na fase de definição do projeto são estabelecidos os objetivos maiores. Esta etapa é comumente chamada de definição do domínio do problema ou do escopo do projeto.

No desenvolvimento de sistemas transacionais, geralmente trabalha-se com aspectos específicos que tratam da automação de determinado procedimento. No projeto de *data warehouse*, inicialmente, deve-se procurar responder a questões gerais tais como: o que deseja-se analisar e porque precisa-se analisar. Respondendo estas questões, consegue-se um entendimento inicial dos requisitos dos usuários e a maneira pela qual estes requisitos serão atendidos.

O principal objetivo em definir o escopo do projeto é prevenir as constantes mudanças durante as fases do ciclo de desenvolvimento à medida que novos requisitos são identificados. No processo de data warehousing, a definição do escopo do projeto requer cuidado especial. Ainda que seja verdadeiro evitar as constantes mudanças à medida que novos requisitos surgem, tem-se o desafio de construir um *data warehouse* flexível e que tenha a habilidade de absorver as consultas ora não conhecidas. Assim, é importante reconhecer que o *data warehouse* final pode resultar em algo um tanto diferente daquele especificado na fase de requisitos (BALLARD et al., 1998).

A fase de definição do projeto envolve o entendimento dos conceitos e tecnologias relacionados ao ambiente de inserção do *data warehouse*, cujo tema é abordado no capítulo 2 desta dissertação. Assim, antes do inicio do processo de construção do *data warehouse* é recomendado que seja realizado um planejamento prévio para determinar a escolha da arquitetura e infraestrutura necessária para possibilitar o pleno desenvolvimento do *data warehouse.*

Um aspecto importante na definição do projeto é a escolha da abordagem de desenvolvimento. A decisão de usar a estratégia *botton up* ou *top down* deve ser tomada com cuidado. Nesta proposta de metodologia adotamos a abordagem de desenvolvimento *botton up.* A construção de ambiente de *data warehouse* corporativo, totalmente integrado, contemplando a cosistência dos dados e aos pré-requisitos de todo os usuários é uma tarefa muito complexa e o risco de fracasso é considerável.

Construir um sistema de *data warehouse* em etapas pequenas incrementais atendendo a grupo específico de usuários torna o projeto menos complexo. Assim, é prudente inciar o projeto selecionando uma área específica atendento a um subconjunto de usuários. O tempo necessário para produzir e aprenstar so primeiros resultados é menor. O usuário começa a familiarizar-se com a nova tecnologia.

### 6.1.3. Modelagem conceitual

Um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um SGBD (Sistema de Gerência Banco de dados). O modelo conceitual registra que dados devem aparecer no banco de dados embora não registre a forma como estes dados são armazenados (HEUSER, 2001).

A fase de modelagem conceitual é o processo pelo qual examina-se a empresa para determinar os tipos de entidades e os relacionamentos entre estas entidades (OZSU & VALURIEZ, 1999). Na modelagem conceitual de *data warehouse* não basta apenas realizar o levantamento de requisitos dos usuários. Adicionalmente, as estruturas dos bancos de dados operacionais devem ser consideradas. Os requisitos dos usuários e as estruturas dos bancos de dados possuem influência estática e dinâmica, caracterizadas pelas possíveis alterações nos requisitos dos usuários e pela mudança na estrutura do banco de dados em questão (BOEHNLEIN et al, 1999).

A maioria dos projetos de *data warehouse* segue uma abordagem evolucionária Inicialmente, o desenvolvimento de um protótipo fornece um conjunto inicial de dados. Posteriormente, o protótipo é alterado e estendido de acordo com o crescimento e mudança dos requerimentos dos usuários. Desta forma, usando a abordagem evolucionária, é importante contemplar a possibilidade da evolução do esquema dimensional. Para garantir a flexibilidade e sua reusabilidade, o esquema dimensional deve ser definido a nível conceitual (SAPIA et al. 1998).

Outro aspecto importante a ser considerado na modelagem conceitual é o fato de que o *data warehouse* envolve toda a organização. Não basta apenas o envolvimento do pessoal do departamento técnico. A estrutura da organização, seus objetivos e metas, a

necessidades de informações gerenciais são aspectos importantes que devem ser considerados nesta fase.

As questões a seguir, sugeridas em (BALLARD et al., 1998) fornecem um auxílio no trabalho de levantamento de requisitos. As respostas a estas questões provavelmente fornecem as informações necessárias ao inicio da modelagem conceitual.

- Quem (Pessoas, grupos, organizações) interessa ao usuário?
- O que (funções) o usuário esta tentando analisar?
- Porque o usuário precisa destes dados?
- Quando (ponto no tempo) os dados precisam ser registrados?
- Onde ocorrem os processos relevantes?
- Como medir a performance ou estado das funções sendo analisadas?

Na figura 6.2 mostramos que existem basicamente duas abordagens para obter os requisitos do *data warehouse* a ser desenvolvido. A primeira alternativa concentra-se mais diretamente no usuário. A segunda alternativa dá maior ênfase aos dados existentes nos sistemas da organização.

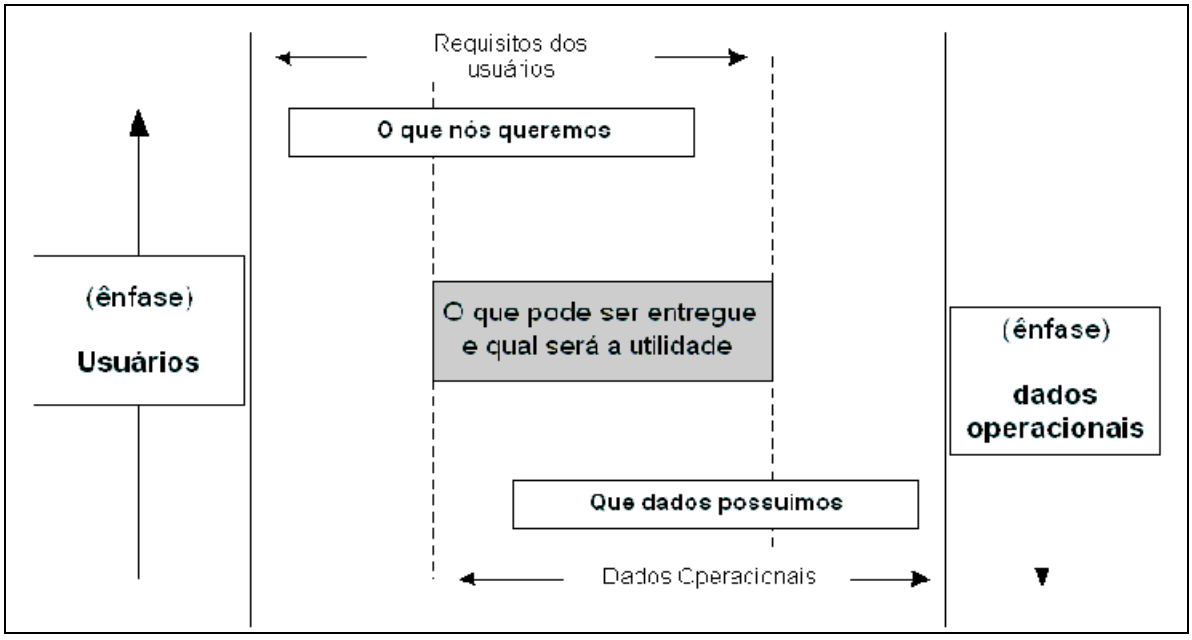



FIGURA 6.2: Duas abordagens para obtenção de requisitos do *data warehouse*

O método de levantamento de requisitos guiado pelo usuário é realizado através da investigação das funções que os usuários executam. Geralmente esse processo se desenvolve através de entrevistas e reuniões. A vantagem deste método é a ênfase naquilo que é necessário para o usuário. Esta abordagem tende a produzir um *data warehouse* útil em um período de tempo menor.

O ponto negativo desta abordagem é o gerenciamento adequado das expectativas dos usuários que precisam ser conscientizados de que possivelmente alguns dos dados que podem ser necessários simplesmente podem não ser disponibilizados num primeiro momento (BALLARD et al., 1998).

O método de obtenção de requisitos a partir dos dados existentes nos bancos de dados da organização é realizado através da análise dos modelos de dados existentes. A maior vantagem desta abordagem é saber desde o inicio que todos os dados podem ser disponibilizados uma vez que o trabalho se limita aos dados que estão disponíveis. Outro benefício é a possibilidade de reduzir o tempo que seria gasto com reuniões e entrevistas com os usuários.

A principal desvantagem desta abordagem reside no risco de estabelecer um conjunto incorreto de requisitos ocasionado pela ausência do usuário no processo. Dependendo do volume de dados e da disponibilidade de modelos de dados esta pode ser tornar-se uma tarefa de alto consumo de tempo.

O resultado da abordagem de levantamento de requisitos com ênfase nos dados disponíveis é prover o usuário com os dados de que se dispõem. Claramente, em dois casos, esta abordagem é apropriada. A primeira é a facilidade de identificar as dimensões de interesse da organização o qual constitui um passo importante na modelagem dimensional. A segunda, através da análise dos dados e relacionamentos, pode-se identificar as áreas de interesse no qual pretende-se focar os esforços de desenvolvimento do *data warehouse*.

Em (TRYFONA et al., 1999) encontra-se uma proposta que extende o modelo E/R. O modelo proposto pelo autor tem o objetivo de auxiliar o projetista na construção de modelos de dados para *data warehouse* contemplando a já conhecida amplamente difundida modelagem E/R. Entretanto, essa extenção ainda não foi incluída nas ferramentas

e produtos disponíveis no mercado. Sem dúvida, seria de grande valia o uso de uma ferramenta *CASE* que auxiliasse na fase da modelagem conceitual. Esperamos que nos próximos lançamentos de novas versões de produtos, estes avanços estejam contemplados.

Também em SAPIA et al. (1998) encontramos uma extensão ao modelo E/R para o paradigma multidimensional. Novos elementos gráficos são introduzidos para estender o modelo E/R conforme mostra a figura 6.3. Nesta metodologia, para a elaboração do modelo conceitual, adota-se a representação gráfica desenvolvida por Sapia.


FIGURA 6.3: Notação gráfica do modelo ME/R

### 6.1.4. Projeto lógico

A etapa de projeto lógico tem o objetivo de transformar o modelo conceitual obtido na fase anterior em um modelo lógico. O modelo lógico define como o banco de dados será implementado em um SGBD específico. Nesta dissertação, o modelo lógico será tratado visando a sua implementação em banco de dados relacional.

Nesta metodologia e para o desenvolvimento do estudo de caso, usa-se a técnica de modelagem dimensional para a criação do projeto lógico do *data warehouse*. De acordo com KIMBALL (1998), a modelagem dimensional é mais apropriada para modelagem de *data warehouse*. Este método foi desenvolvido a partir de observações de casos práticos e tem sido a abordagem dominante representando grande avanço no processo de modelagem de *data warehouse* e *data mart*s.

A modelagem dimensional é caracterizada pelo esquema estrela e é usado para representar uma implementação em um banco de dados relacional. O esquema estrela consiste de uma tabela de fatos (central) rodeada de tabelas de dimensão. A tabela de fatos contém os elementos mensuráveis do negócio a ser analisado. Cada fato é analisado através das dimensões que o compõem (GATZIU & VAVOURAS, 2000).

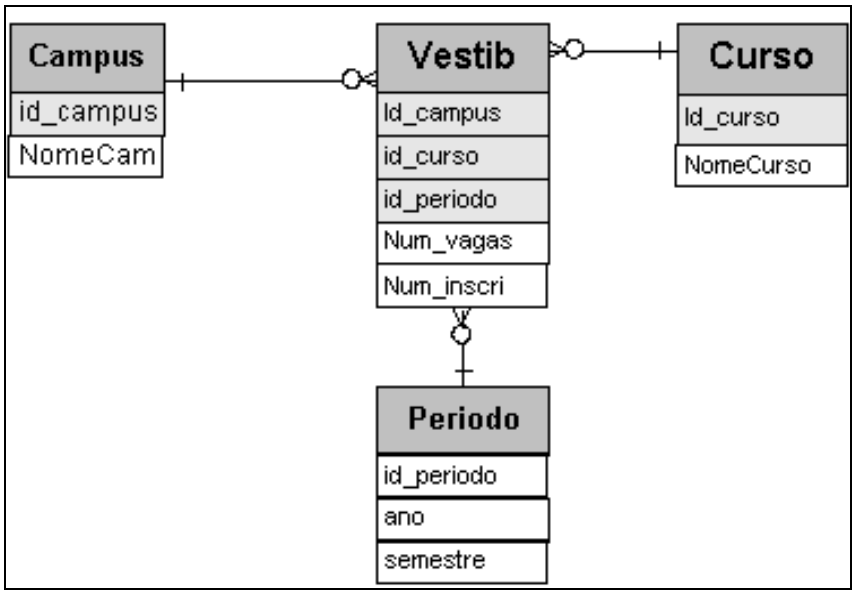



FIGURA 6.4: Um modelo dimensional típico

A figura 6.4 mostra o modelo dimensional típico produzido na etapa de projeto lógico. Este modelo deve ter a capacidade de representar os requisitos obtidos na fase da modelagem conceitual. Os requisitos possuem dois elementos fundamentais: O que está sendo analisado e qual o critério de avaliação para aquilo que está sendo analisado. Desta forma podemos referenciar o critério como as medidas e o que está sendo analisado como as dimensões (BALLARD et al., 1998). O primeiro passo para a criação do modelo dimensional é a identificação dos fatos e das medidas a partir do nosso conjunto de requisitos.

O modelo dimensional da figura 6.4 representa um exemplo simples de um sistema de vestibular. No centro da figura encontra-se a tabela de fatos (VESTIB) o qual registra o número de inscritos no vestibular e o número de vagas. Neste exemplo são consideradas três dimensões (período, campus, curso). A tabela de fatos possui dois atributos que medem o fato. O atributo *num_vagas* que representa o número de vagas disponíveis no curso,

naquele campus no período específico. Por exemplo, O curso de **administração** oferece 50 vagas no campus **Santa Rosa** no vestibular referente ao **primeiro semestre de 2002.** O atributo *num_inscr* registra o número de candidatos inscritos nas dimensões especificadas. A partir destes atributos pode-se calcular, por exemplo, a ociosidade, a relação candidato vaga e os cursos mais procurados pelos candidatos.

Outro detalhe a ser observado no exemplo da figura 6.4 é a definição do nível de granularidade semestral. Isto significa que os dados são agregados semestralmente por curso e campus. A decisão sobre a determinação do nível de granularidade deve estar fundamentada nos requisitos estabelecidos na fase de modelagem conceitual.

### 6.1.5. Projeto físico

Há diferentes formas de se realizar o projeto físico do banco de dados. Esta função é dedicada ao Administrador de banco de dados (DBA). Com o advento do *data warehouse*, novos aspectos técnicos devem ser considerados na implementação física do modelo de dados (SINGH, 2001). Os principais aspectos a serem considerados no projeto físico do *data warehouse* são:

- Indexação;
- Materialização de visões;
- Particionamento, paralelismo;
- Nível de redundância dos dados
- Sintonia dos parâmetros do banco de dados.

Em (HERDEM, 2000) encontramos uma ótima fonte para pesquisa para cada um dos tópicos acima citados. Entretanto, nesta dissertação, os aspectos relacionados à implementação física do banco de dados não serão abordados em maiores detalhes. A etapa mais importante está relacionada à geração dos esquemas físicos de dados os quais serão utilizados para a criação das estruturas de dados. O estudo de caso desta dissertação será implementado em base de dados relacional. Assim, serão utilizados todo o conhecimento e

experiência adquirida pela equipe técnica de administração de banco de dados da instituição.

A sintonia do banco de dados é fundamental no ambiente de *data warehouse* dado que a natureza da carga é diferente do ambiente transacional. Os ambiente OLTP são configurados para realizar as transações dos vários usuários simultâneos no menor tempo possível. Os parâmetros de configuração ajustados são responsáveis pela melhora do desempenho em 20 a 25% (HAYES & GUNNING, 2002). Os 75% restantes derivam do ajustes dos comandos (SQL) de consulta. Isso envolve alterações no projeto físico do banco de dados, disponibilidade e características dos índices, replicação e particionamento de tabelas.

## 6.2. Outros aspectos importantes da metodologia proposta

Na metodologia de desenvolvimento proposta ainda faltam aspectos que assumem um papel fundamental na construção de um *data warehouse*. A seguir, discute-se brevemente estes aspectos uma vez que não é objetivo desta dissertação tratar estes tópicos com maiores detalhes.

### 6.2.1. Metadados

Os metadados são dados sobre dados. Os metadados mantêm informações sobre o conteúdo que está armazenado no *data warehouse*. Na comunidade de usuários, normalmente, na há um alto grau de especialização. Assim, o usuário para usar eficientemente o *data warehouse*, utiliza os metadados para efetuar o seu processamento analítico (INMON, 1997).

Em COME (1999) encontramos um bom referencial sobre a importância dos metadados em um *Data warehouse*. O autor cita algumas perguntas básicas que também devem ser respondidas pelos metadados:

- Quais tabelas, atributos e chaves o *Data warehouse* contém?

- Qual é a origem de cada conjunto de dados?
- Que transformação lógica foi usada na carga do dado?
- Como o metadado tem mudado ao longo do tempo?
- Quais *aliases* existem e como eles se relacionam?
- Quais são as referências-cruzadas entre termos técnicos e de negócios?
- Com qual freqüência os dados são carregados?
- Qual o volume de dados existente?

Geralmente os metadados são esquecidos na promessa de serem criados ao final do projeto. A experiência no desenvolvimento de sistemas transacionais mostra que a fase de documentação geralmente era esquecida pelo desenvolvedor. No entanto os projetos de *data warehouse* que não tem preocupação com os metadados não tiveram caráter prático. O usuário não usa, não adere. Os usuários não confiam nos números. Esta situação é revertida, mostrando através dos metadados, de onde vem os números e de que forma foram calculados ou gerados (TANNEMBAUM, 2002). Os metadados são elaborados gradativamente ao longo de todo o processo de desenvolvimento do *data warehouse*.

MARCO (2001) indica que uma solução de metadados deve incluir:

- Os objetivos técnicos e de negócios do repositório de dados;
- Uma ferramenta para a sua administração;
- Pessoal qualificado para a elaboração de cada tarefa (modelagem, desenvolvimento, manutenção);
- Proporcionar acesso a todos os usuários conforme as políticas de segurança.

### 6.2.2. Granularidade

A granularidade do *data warehouse* registra que nível de detalhe os dados estarão disponíveis para o análise do usuário, isto é, determina a sua dimensionalidade possuindo influência direta no tamanho do banco de dados. A decisão sobre a granularidade dos dados

é um dos aspectos mais importantes na construção do *data warehouse*. A escolha de um nível de granularidade inadequada pode comprometer e até inviabilizar o uso do *data warehouse*.

O grau de granularidade do *data warehouse* não é definido a partir de uma regra exata ou fixa. A definição da granularidade é específica de cada projeto. Ela é determinada a partir do grau de detalhamento que deseja obter na consulta aos dados. Por exemplo, em um sistema de vendas, pode-se agrupar o total das vendas por dia, semana, mês, bimestre, trimestre, semestre e ano. Em determinadas situações, pode ser desejável implementar um nível dual de granularidade (INMON,1997).

### 6.2.3. Atualização do *data warehouse*

Um aspecto importante a ser considerado em um projeto de *data warehouse* é o planejamento da periodicidade de atulização dos dados. Uma vez construído, o *data warehouse* recebe a primeira carga de dados. As sucessivas cargas devem então ser planejadas considerando apenas as modificações do ambiente operacional no intervalo considerado.

A cada ciclo de atualização dos dados, ocorre o processo de extração, transformação e carga dos dados integrando-os no *data warehouse.* Este é o processo que demanda o maior volume de trabalho e consumo de tempo de todo o projeto. A partir da primeira carga, este processo se repete periodicamente sendo fundamental o estabelecimento de uma rotina de execução bem definida.

A etapa de atualização de dados deve ser suportada pela ferramenta de desenvolvimento *data warehouse* a qual deve suportar as seguintes atividades:

- Automação do processo de extração, conversão e carga dos dados;
- Definição da periodicidade da atualização;
- Possibilidade de integração com outras ferramentas.

## 6.3. Considerações finais

Este capítulo apresentou uma proposta de metodologia de desenvolvimento de *data warehouse*. O objetivo principal foi detalhar os aspectos não contemplados pelas metodologias apresentadas anteriormente com destaque para as fases modelagem conceitual e projeto lógico. No próximo capítulo será desenvolvido um estudo de caso no qual aplicamos a metodologia descrita neste capítulo. O objetivo do estudo de caso é avaliar a aplicabilidade e viabilidade da metodologia aqui apresentada. O estudo de caso usa como exemplo, o sistema de concurso vestibular da UNIJUÍ – Universidade Regional do Noroeste do Estado do Rio Grande do Sul.

# 7. ESTUDO DE CASO

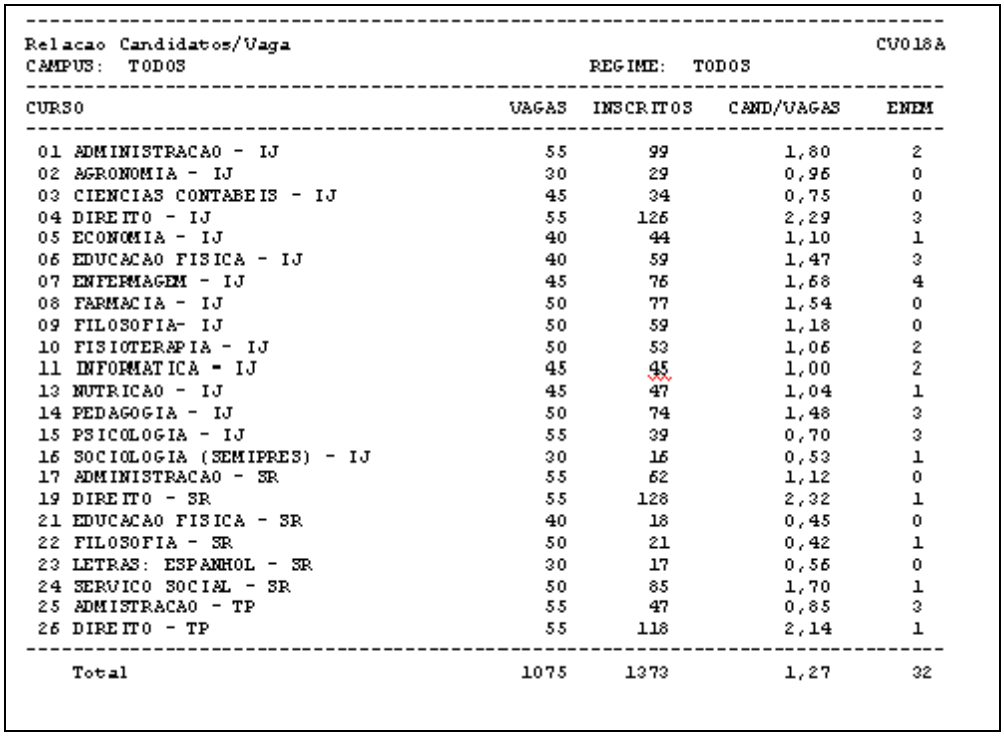
Este capítulo desenvolve um estudo de caso envolvendo o projeto e desenvolvimento de um *data warehouse* utilizando a metodologia proposta no capítulo anterior. O objetivo deste estudo de caso é avaliar e verificar a aplicabilidade da metodologia proposta. Inicialmente aborda-se o sistema atual de geração de informações gerenciais tomando como exemplo os dados do sistema de concurso vestibular da UNIJUI. Na seqüência, desenvolvemos o estudo de caso aplicando as várias fases da metodologia.

## 7.1. O sistema atual de geração dos dados estatísticos

Antes de iniciar o estudo de caso através do desenvolvimento do *data warehouse*, vamos apresentar, de forma bastante sintética, o sistema atual de geração de dados consolidados e que fornecem as informações desejadas pelas instâncias administrativas da UNIJUÍ. A título de exemplo, usaremos o sistema de vestibular para demonstrar o processo. No entanto, isso se aplica aos demais sistemas que compõem a publicação periódica que reúne informações sobre diversos aspectos da vida acadêmica da UNIJUÍ.

O sistema de vestibular foi desenvolvido no primeiro semestre de 1992. É um sistema muito confiável e gerencia todo o concurso vestibular o qual acontece duas vezes ao ano. O vestibular de verão que acontece, na grande maioria das vezes, no mês de janeiro e o de inverno realizado no mês de julho de cada ano.

Finalizado o processo de vestibular, o analista responsável pelo sistema de vestibular gera uma série de relatórios que fornecem os dados estatísticos acerca do concurso vestibular em questão. Estes relatórios são então encaminhados a um funcionário da vice-reitoria de administração. A figura 7.1 mostra um exemplo deste relatório.



Relacao Candidatos/Vaga — CV018A
CAMPUS: TODOS — REGIME: TODOS

| CURSO | VAGAS | INSCRITOS | CAND/VAGAS | ENEM |
|---|---|---|---|---|
| 01 ADMINISTRACAO - IJ | 55 | 99 | 1,80 | 2 |
| 02 AGRONOMIA - IJ | 30 | 29 | 0,96 | 0 |
| 03 CIENCIAS CONTABEIS - IJ | 45 | 34 | 0,75 | 0 |
| 04 DIREITO - IJ | 55 | 126 | 2,29 | 3 |
| 05 ECONOMIA - IJ | 40 | 44 | 1,10 | 1 |
| 06 EDUCACAO FISICA - IJ | 40 | 59 | 1,47 | 3 |
| 07 ENFERMAGEM - IJ | 45 | 76 | 1,68 | 4 |
| 08 FARMACIA - IJ | 50 | 77 | 1,54 | 0 |
| 09 FILOSOFIA- IJ | 50 | 59 | 1,18 | 0 |
| 10 FISIOTERAPIA - IJ | 50 | 53 | 1,06 | 2 |
| 11 INFORMATICA - IJ | 45 | 45 | 1,00 | 2 |
| 13 NUTRICAO - IJ | 45 | 47 | 1,04 | 1 |
| 14 PEDAGOGIA - IJ | 50 | 74 | 1,48 | 3 |
| 15 PSICOLOGIA - IJ | 55 | 39 | 0,70 | 3 |
| 16 SOCIOLOGIA (SEMIPRES) - IJ | 30 | 16 | 0,53 | 1 |
| 17 ADMINISTRACAO - SR | 55 | 62 | 1,12 | 0 |
| 19 DIREITO - SR | 55 | 128 | 2,32 | 1 |
| 21 EDUCACAO FISICA - SR | 40 | 18 | 0,45 | 0 |
| 22 FILOSOFIA - SR | 50 | 21 | 0,42 | 1 |
| 23 LETRAS: ESPANHOL - SR | 30 | 17 | 0,56 | 0 |
| 24 SERVICO SOCIAL - SR | 50 | 85 | 1,70 | 1 |
| 25 ADMISTRACAO - TP | 55 | 47 | 0,85 | 3 |
| 26 DIREITO - TP | 55 | 118 | 2,14 | 1 |
| Total | 1075 | 1373 | 1,27 | 32 |

FIGURA 7.1: Exemplo de relatório estatístico

A partir dos vários relatórios gerados pelo sistema, a vice-reitoria de administração tabula estes dados em formato que facilite o trabalho de análise. Esta tabulação resulta em uma série de tabelas organizadas de acordo com o requisito de informação desejado. Algumas das tabelas geradas no processo de tabulação dos dados são:

- Número de vagas oferecidas pelos cursos de graduação na UNIJUÍ
- Número de candidatos inscritos no vestibular para os cursos de graduação da UNIJUÍ
- Número de candidatos aprovados no vestibular para os cursos de graduação da UNIJUÍ

- Número de vestibulandos que realizaram matrículas (ingresso na UNIJUÍ)
- Relação candidato/vaga no vestibular da UNIJUÍ
- Percentual de utilização das vagas oferecidas no vestibular.

A tabela 7.1 mostra um exemplo de uma tabela gerada através deste processo, no caso, o quadro de vagas oferecidas no vestibular pelos cursos de graduação da UNIJUI. Esta é uma visão parcial da planilha e serve apenas para ilustrar o exemplo. Neste exemplo, estão sendo considerados apenas alguns cursos no período de 1992 a 1996, do campus Ijuí, regime regular.

| VAGAS OFERECIDAS NO VESTIBULAR PELOS CURSOS DE GRADUAÇÃO DA UNIJUÍ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **CURSOS/REGIME/CAMPUS** | **92/1** | **92/2** | **93/1** | **93/2** | **94/1** | **94/2** | **95/1** | **95/2** | **96/1** | **96/2** |
| **CAMPUS IJUÍ** | 1270 | 340 | 1420 | 215 | 1430 | 255 | 1550 | 478 | 1395 | 365 |
| **REGIME REGULAR** | 1030 | 340 | 1140 | 215 | 1100 | 255 | 1220 | 478 | 1115 | 365 |
| Administração | 60 | 60 | 60 | 60 | 60 | 60 | 55 | 55 | 55 | 55 |
| Agronomia | 55 | | 55 | | 55 | | 55 | | 55 | 55 |
| Ciências | 90 | | 90 | | 90 | | 90 | | 60 | |
| Ciências Contábeis | 120 | | 60 | | 60 | | 60 | 55 | 60 | 60 |
| Comunicação Social | | | | | | | | 90 | 90 | |
| Design | | | | | | | | | | |
| Direito | 110 | 55 | 110 | 55 | 110 | 55 | 110 | 55 | 110 | 55 |
| Economia | 55 | | 55 | | 55 | | 55 | | 55 | |

TABELA 7.1: Exemplo de planilha com dados estatísticos

Este sistema de geração de dados estatísticos tem algumas dificuldades e limita o usuário no processo de análise das informações, destacando-se:

- O processo todo é manual, dispendioso e propenso a erros;
- Exige re-trabalho, pois a planilhas são digitadas a partir dos relatórios;
- A planilha provê uma capacidade muito limitada de análise;

- O usuário, para obter as informações, muitas vezes deve manipular e analisar várias planilhas;
- A planilha eletrônica não fornece suporte adequado para documentação explicando a maneira como esses resultados foram calculados;
- Dificuldade para ampliar as dimensões da análise dos dados

## 7.2. O sistema novo utilizando *data warehouse*

No estudo de caso desta dissertação pretende-se, através da tecnologia de *data warehousing*, propor uma alternativa para a geração dos dados estatísticos e que elimine as dificuldades e limitações encontradas no sistema atual. Através do desenvolvimento deste estudo de caso, teremos a oportunidade de adquirir maior experiência nesta área. Caso, ao final deste trabalho, esta solução realmente tornar-se uma alternativa viável, poderá ser expandida e consolidada integralmente ao sistema de informações da instituição. O estudo de caso será desenvolvido utilizando a metodologia descrita no capítulo 6.

### 7.2.1. Gerência do projeto

A primeira etapa de um projeto de *data warehouse* é estabelecer o componente de gerência do projeto. Chama-se a atenção que a gerência neste nível corresponde ao projeto e não a gerência do *data warehouse*. A gerência do projeto deve ter prazo finito enquanto que a gerência do *data warehouse* é contínua e preocupa-se com a execução dos processos de *data warehousing*. A execução destes processos é uma atividade contínua durante todo o ciclo de vida do *data warehouse*.

O componente de gerência do projeto têm a responsabilidade de estabelecer o plano geral do projeto. Este plano deve ser conhecido por todos os membros que farão parte da equipe de desenvolvimento do projeto. O plano deve estabelecer o prazo do projeto, os recursos disponíveis e principalmente a expectativa dos usuários com relação ao projeto ora sendo iniciado.

O maior desafio desta etapa é a identificação do responsável pela gerência do projeto. O gerente de projeto tem a responsabilidade de estabelecer as pricipais variáveis do projeto incluindo:

- As funções que o *data warehouse* irá disponibilizar;
- Alocação de recursos (máquinas, ferramentas, pessoas);
- Qualidade (definição de prazos não realísticos podem levar a equipe a seguir atalhos e comprometer a qualidade do *data warehouse).*

### 7.2.2. Definição do projeto

Neste estudo de caso será desenvolvido um sistema automatizado que será apresentado como alternativa ao sistema atual de geração de dados analíticos eliminando as atuais dificuldades e limitações apresentadas anteriormente. Neste estudo de caso, serão utilizadas ferramentas de desenvolvimento de *data warehouse* as quais auxiliam o projetista nas várias fases do projeto. Estas ferramentas estão divididas nas seguintes categorias:

- Servidor de banco de dados: IBM DB2 V7.2
- Ferramenta de *Data warehouse*: IBM DB2 *Warehouse Manager* V7.2
- Ferramenta OLAP: IBM DB2 OLAP *STARTER KIT* V7.2

O produto IBM DB2 *Warehouse Manager* também oferece um conjunto de recursos que auxiliam na criação dos metadados ao longo do desenvolvimento do *data warehouse*.

Para viabilizar o estudo de caso, usaremos o banco de dados operacional da UNIJUI. Especificamente, o escopo do estudo de caso limita-se ao sistema de concurso vestibular. Para implementar o *data warehouse* objeto do nosso estudo será utilizada a arquitetura centralizada. Esta solução poderá posteriormente ser expandida através da incorporação de outros módulos.

### 7.2.3. Modelagem conceitual

Neste estudo de caso, a fase de levantamento de requisitos teve como base às informações do usuário e os dados existentes nos sistemas operacionais. O estudo do sistema atual (conforme descrito no item 7.1) foi de grande importância à compreensão dos requisitos do sistema a ser projetado.

A figura 7.2 mostra o modelo conceitual elaborado a partir dos requisitos levantados. No centro do esquema encontramos o fato vestibular que é composto por seis medidas:

- **Vagas**: Corresponde ao número de vagas oferecidas
- **Inscritos**: Número de candidatos inscritos
- **Classificados**: Número de candidatos classificados no vestibular
- **Aprovados**: Número de candidatos aprovados no vestibular
- **Não aprovados**: Número de candidatos não aprovados no vestibular
- **Suplentes**: Número de candidatos suplentes.

Além da tabela de fatos, o esquema conceitual possui cincos dimensões:

- A dimensão **CAMPUS**: A Universidade oferece curso de graduação em vários campus.
- A dimensão **REGIME**: Um curso pode pertencer ao Regime Regular (norma) ou Especial (período de férias, meses de janeiro, fevereiro e julho).
- A dimensão **CURSO**: Os vários curso oferecidos a cada vestibular.
- A dimensão **TEMPO**: A cada ano são realizados dois vestibulares (Semestral).
- A dimensão **CIDADE**: Tem a finalidade de realizar a estatística da origem dos candidatos do vestibular.

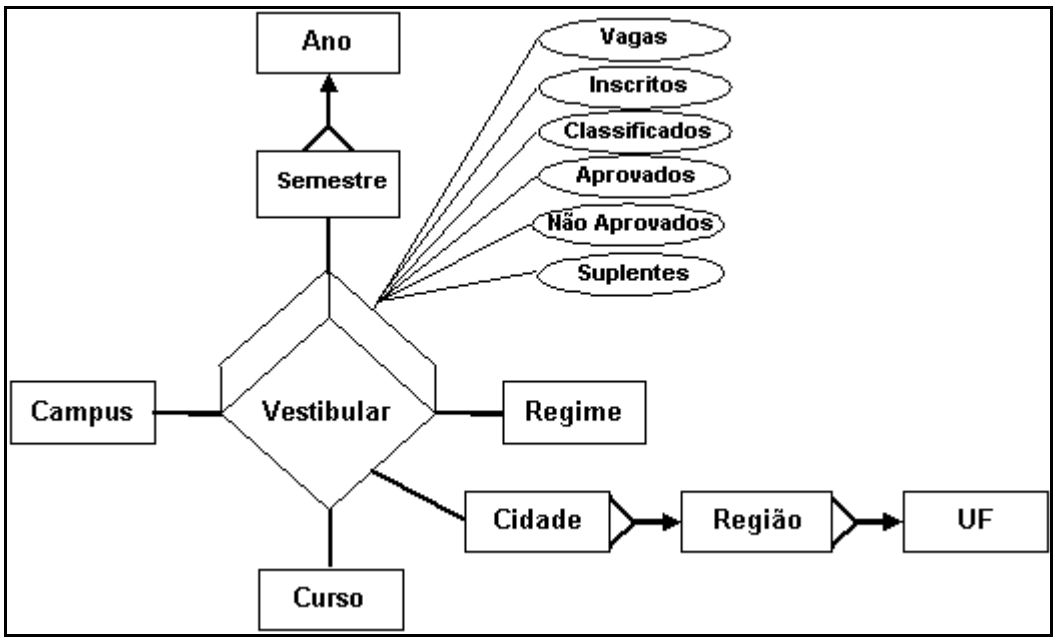



FIGURA 7.2: Modelo conceitual do sistema de vestibular

### 7.2.4. Projeto lógico

A fase de projeto lógico tem o objetivo de elaborar o esquema estrela do *data warehouse*. Esta fase é inteiramente desenvolvida através da utilização de uma ferramenta que suporta a construção do esquema estrela. O Centro de *Data warehouse* do IBM DB2 guia o projetista através das várias etapas do projeto lógico conforme mostrado na figura 7.3.

A primeira etapa do projeto lógico do *data warehouse* é a definição de um assunto. Um assunto compreende um conjunto de processos relacionados á uma área especifica do negócio. No presente estudo de caso, foi definido o assunto Vestibular. O objetivo principal de um assunto é a elaboração de um esquema de *data warehouse* (esquema estrela). Este esquema é construído gradativamente através dos processos que estão relacionados ao assunto.

Um processo tem a finalidade de transformar os dados que estão armazenados nos sistemas fonte. No centro de *data warehouse*, as fontes de dados são identificadas no grupo

*warehouse sources*, conforme mostra a figura 7.3. Cabe ressaltar que a origem dos dados pode derivar de várias bases de dados e podem estar armazenadas em sistemas diferentes.

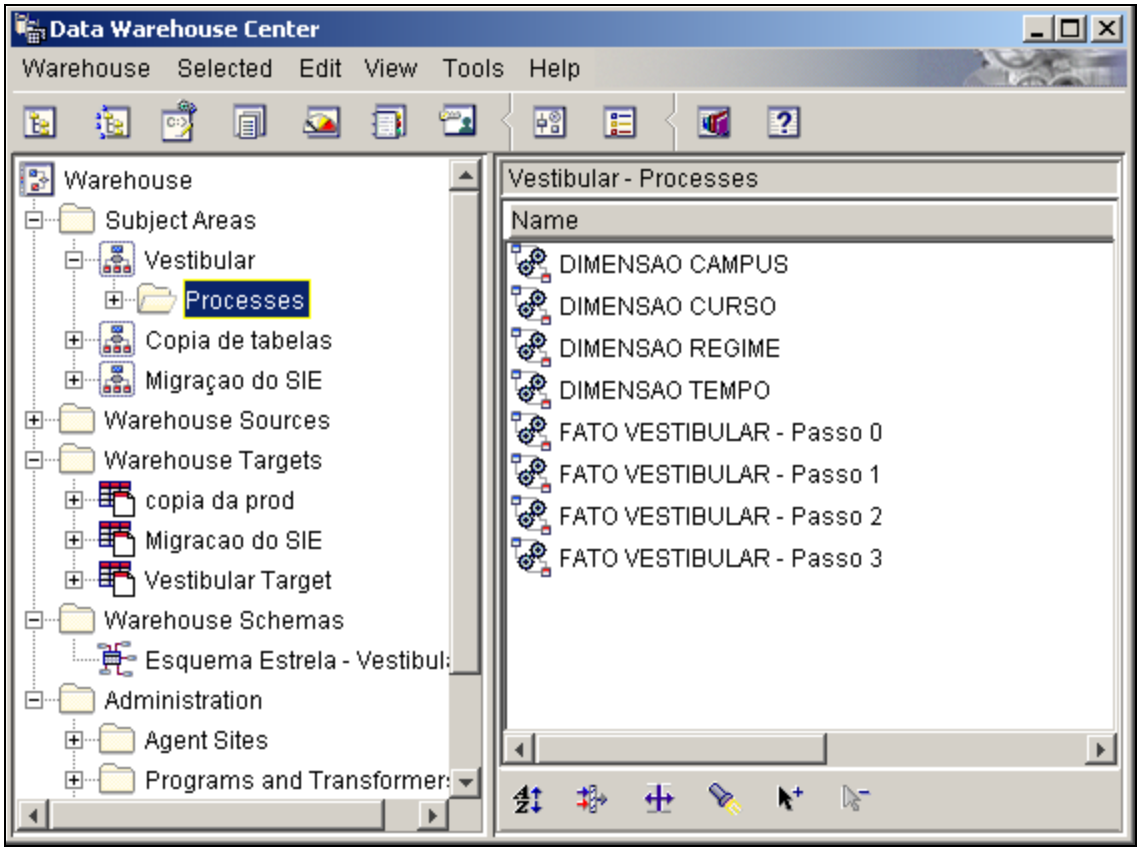



FIGURA 7.3: Centro de *data warehouse* do IBM DB2.

O processo de transformação tem o objetivo de transformar os dados fontes em um formato específico apropriado para ser explorado através das ferramentas OLAP. As tabelas resultantes do processo de transformação são armazenados no destino do *data warehouse* (*Warehouse Targets*).

Um exemplo de processo de transformação de dados é mostrado na figura 7.4. Neste exemplo, o processo transforma os dados de um arquivo texto para uma tabela relacional. A tabela relacional resultante é armazenada no banco de dados do *data warehouse* e compreende a dimensão Tempo do esquema estrela resultante.

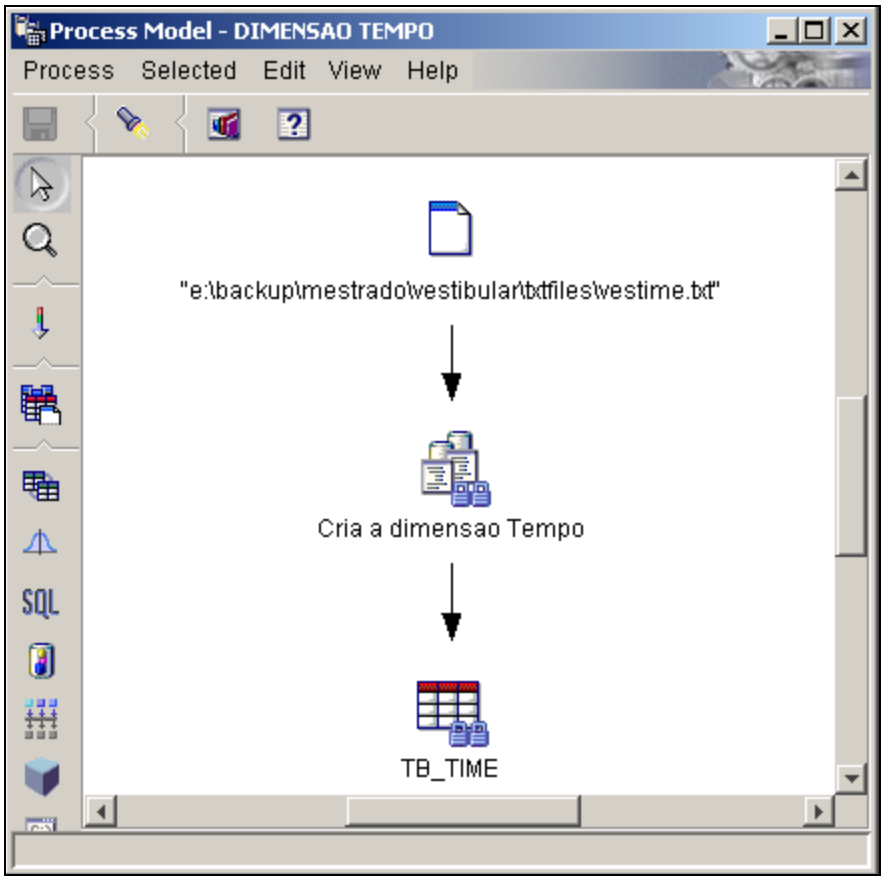



FIGURA 7.4: Processo de criação da dimensão Tempo

Através do uso do centro de *data warehouse*, podemos modelar complexos processos de transformação de dados. Este trabalho não tem o objetivo de mostrar as potencialidades (e limitações) do centro de *data warehouse*. Apenas serão apresentados aqueles recursos utilizados na fase de criação do estudo de caso.

As outras dimensões pertencentes ao esquema estrela foram transformadas por um processo semelhante ao descrito anteriormente.

A figura 7.5 mostra o processo que cria a tabela de fatos do esquema estrela. A tabela é criada a partir de uma série de processos que são executados em seqüência. O resultado do processo de transformação é a criação de uma tabela que reúne os atributos que compreendem as medidas do fato. Neste exemplo, os atributos compreendem a quantidade de alunos inscritos, classificados, aprovados, não aprovados e suplentes. O processo poderia ser simplificado através de um *Join* único entre as tabelas origem. Esta

opção foi adotada para exemplificar o uso de vários passos e a possibilidade de uso de tabelas temporárias no processo de transformação dos dados que é bastante comum em ambiente de *data warehouse*.

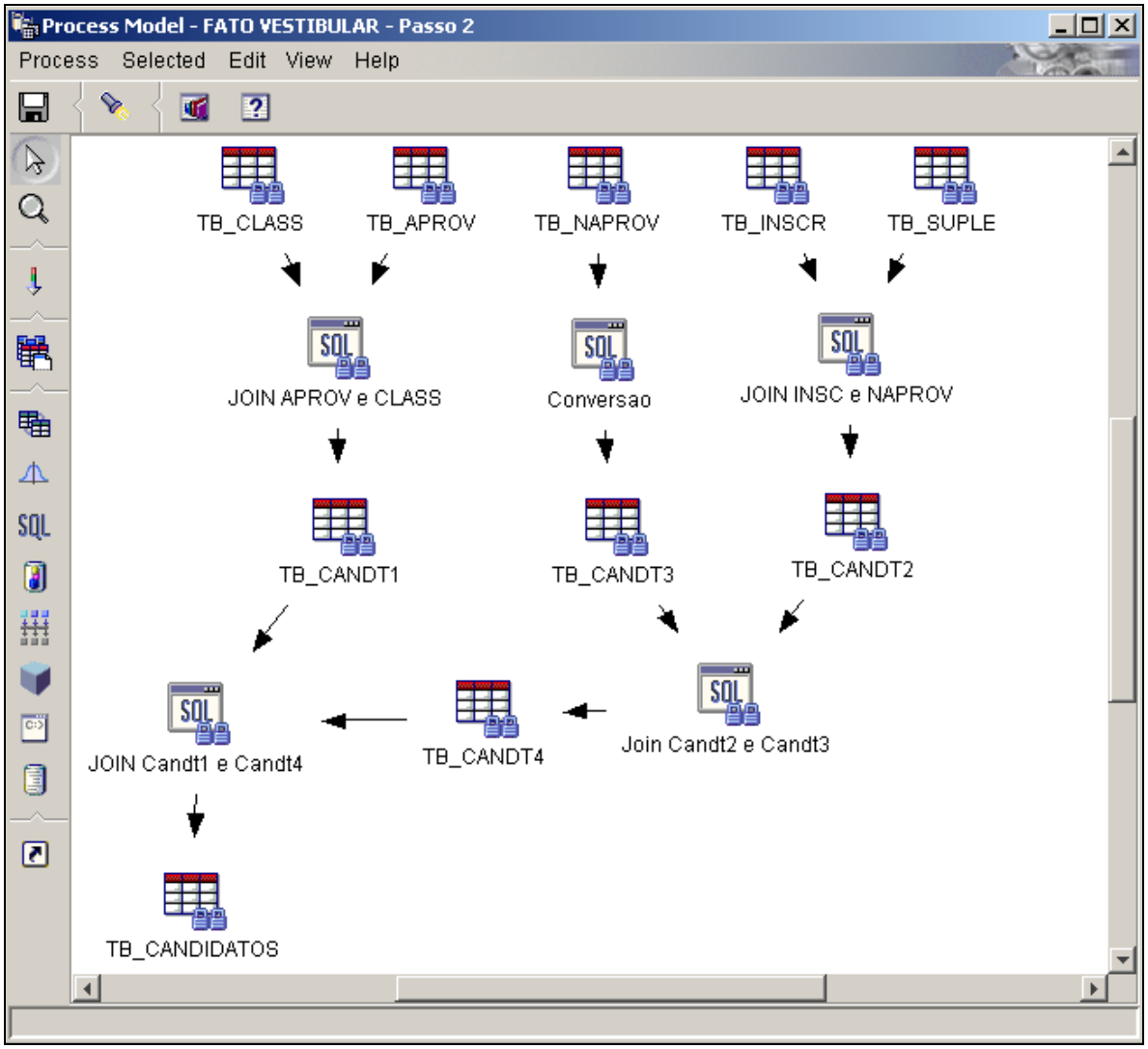



FIGURA 7.5: Processo de criação da tabela de Fatos do vestibular

Ao final da execução de todos os processos de transformação dos dados fonte, obtemos então o esquema estrela resultante, conforme mostrado na figura 7.6. Este esquema será então utilizado posteriormente por uma ferramenta OLAP que será utilizada para a construção de aplicações que serão utilizadas pelos usuários do *data warehouse*.

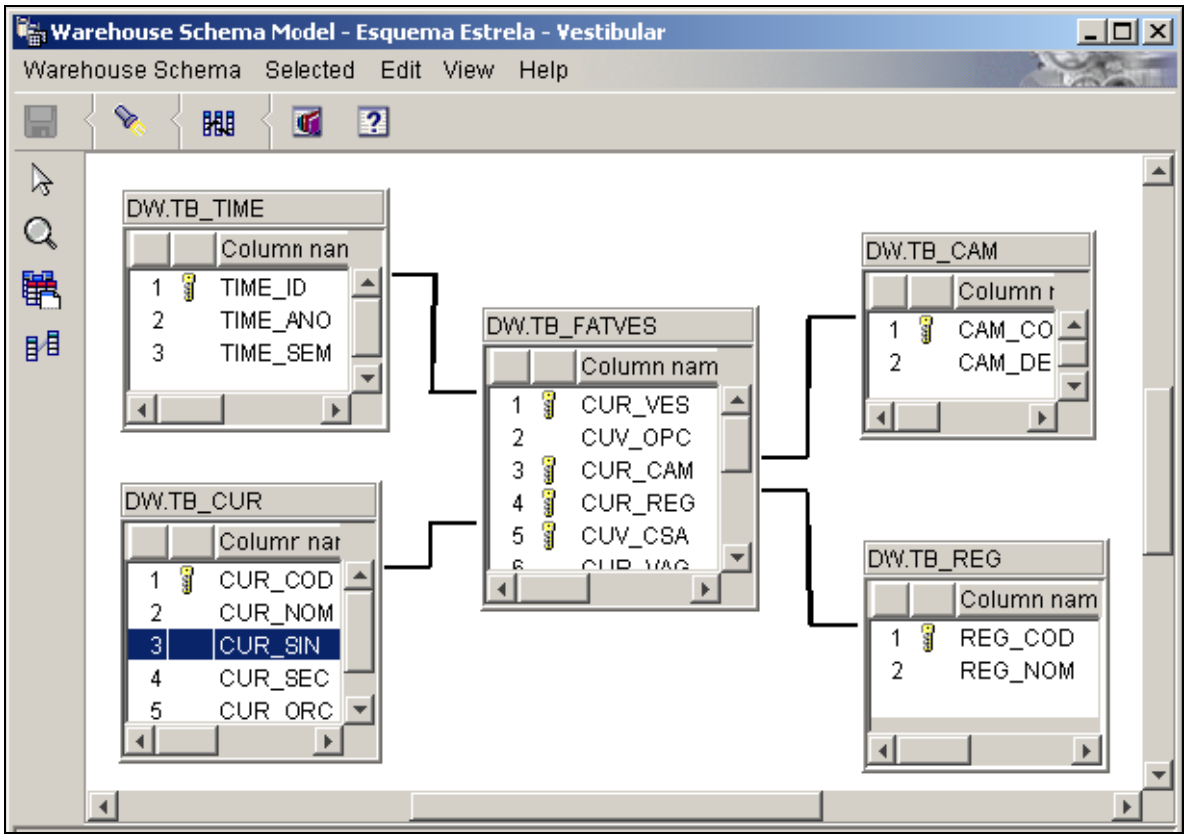



FIGURA 7.6: Esquema estrela do sistema de vestibular

O esquema estrela da figura acima compreende as seguintes tabelas:

Uma tabela de fatos (DW.TB_FATVES)

Quatro tabelas dimensionais:

Dimensão Campus (DW.TB_CAM);

Dimensão Regime (DW.TB_REG);

Dimensão Curso (DW.TB_CUR);

Dimensão Tempo (DW.TB_TIME).

### 7.2.5. Projeto físico

O projeto físico do banco de dados é dependente da arquitetura do SGBD onde o *data warehouse* será implementado. Neste estudo de caso, a implementação será efetivada em um SGBD relacional. No desenvolvimento do projeto lógico, a ferramenta de desenvolvimento utizada automaticamente cria o esquema físico do banco de dados (tabelas, índices). Entretanto, considerando as diferenças de um ambiente de banco de dados tradicional e um ambiente de *data warehouse*, alguns aspectos com relação ao projeto físico devem ser considerados:

- Particionamento;
- Materialização de consultas;
- Paralelismo;
- Parâmetros de configuração do SGBD;
- Indexação.

Cada tópico enumerado acima deve ser detalhado para identificar os fatores que podem melhorar a qualidade da solução sendo implementada. Este é um trabalho contínuo de monitoração e ajuste do banco de dados que deve realizado por profissional competente e que conheça as características do ambiente no qual a solução está sendo implementada.

## 7.3. Outros aspectos importantes

A seguir apresentamos outros aspectos importantes que devem ser considerados em um projeto de *data warehouse*. Estes aspectos não são específicos de determinada fase. Entretanto, assumem importância fundamental em projeto bem sucedido de *data warehouse*.

### 7.3.1. Granularidade

Um aspecto importante no desenvolvimento do *data warehouse* é a definição do nível de granularidade. Neste estudo de caso foi definido o nível de granularidade semestral. Ou seja, os dados são agrupados semestralmente através das dimensões Campus, Regime e Curso.

### 7.3.2. Metadados

Os metadados foram elaborados gradativamente ao longo do processo de criação do *data warehouse*. À medida que cada passo foi sendo criado, os metadados também foram elaborados. Ao final do projeto, os metadados são exportados para que possam ser utilizados pelos usuários. O Metadados podem ser explorados através de uma ferramenta chamada *Information Catalog Manager* conforme mostra a figura 7.7. Esta ferramenta está incluída no Centro de *data warehouse* do IBM DB2.

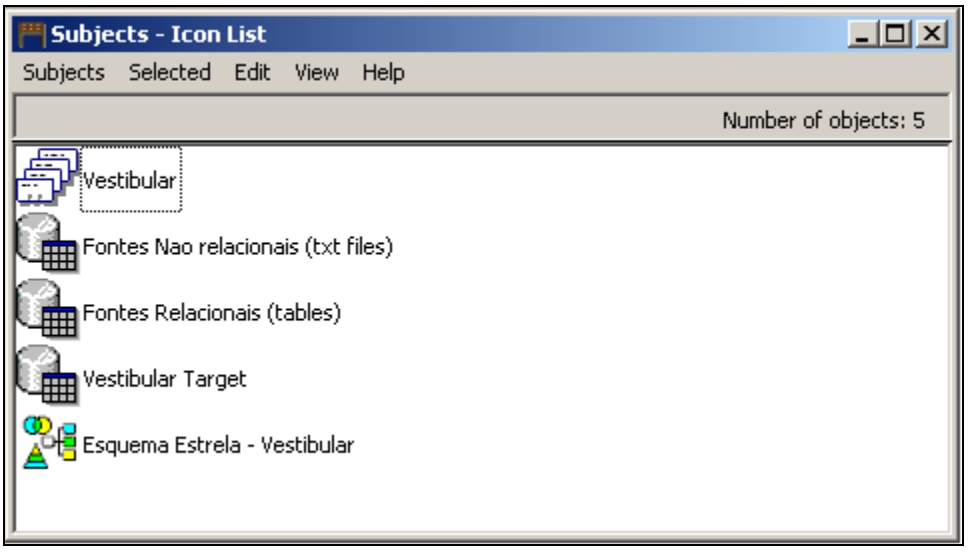



FIGURA 7.7: Ferramenta para exploração dos metadados

Cada ícone da figura representa um grupo de informações que podem ser exploradas pelo usuário. Ao clicar no ícone vestibular, por exemplo, a ferramenta abre todos os processos que fazem parte deste assunto. O ícone fontes relacionais descreve todas as tabelas fonte que foram utilizadas no processo de criação do *data warehouse*. Através da exploração dos metadados, os usuários podem encontrar as tabelas que originaram os dados do *data warehouse* (IBM, 2000).

### 7.3.3. Atualização do *Data Warehouse*

A etapa de atualização de dados deve ser suportada pela ferramenta de desenvolvimento *data warehouse* a qual deve suportar as seguintes atividades:

- Automação do processo de extração, conversão e carga dos dados;
- Definição da periodicidade da atualização;
- Possibilidade de integração com outras ferramentas.

## 7.4. Considerações finais

Este capítulo desenvolveu um estudo de caso envolvendo a construção de um *data warehouse* utilizando a metodologia proposta no capítulo anterior. O *data warehouse* foi desenvolvido aplicando-se as várias fases da metodologia proposta no capítulo anterior. Inicialmente, foram definidos o escopo do projeto e o levantamento dos requisistos do sistema. Segue-se o estudo de caso com a modelagem conceitual, projeto lógico e físico do *data warehouse*. Os metadados são desenvolvidos conjuntamento durante as várias fases do projeto. Concluí-se que, o desenvolvimento do data warehouse através da aplicação das etapas da metodologia proposta, torna o processo mais sistemático reduzindo a chances de fracasso do projeto. Concluí-se também que a metodologia mostrou-se eficaz e de provada aplicabilidade prática podendo ser utilizada integralmente em projetos de sistemas de *data warehouse*.

# 8. CONCLUSÃO

## 8.1. Considerações gerais

Este trabalho teve dois objetivos básicos. O primeiro objetivo foi, através de pesquisa bibliográfica, identificar uma metodologia para desenvolvimento de *data warehouse*. O segundo objetivo era desenvolver um estudo de caso de projeto de *data warehouse* aplicando a metodologia identificada avaliando a sua aplicabilidade.

O trabalho de pesquisa bibliográfica resultou na identificação de três metologias as quais são apresentadas no capítulo 5. Entretanto, estas três metodologias mostraram-se insuficientes para serem utilizadas como referência para o estudo de caso. As razões e limitações desta insuficiência são apresentadas ao final do capítulo 5. Optou-se então, a partir das idéias dos autores das metodologias apresentadas, propor uma nova metodologia que fosse mais abrangente e detalhada em comparação com as metodologias apresentadas. Assim no capítulo 6, apresentamos essa proposta de metodologia que foi complementada a partir de obras de outros autores. No capítulo 7, através de um estudo, avaliamos a aplicabilidade da metodologia apresentada. Concluímos que a proposta elaborada constitui um avanço em relação as anteriores pois apresenta uma sistemática mais apropriada a qual adere à realidade dos sistemas existentes nas empresas. Também, valoriza a experiência da equipe no desenvolvimento de sistemas transacionais pois as fases que compõem a metodologia já são largamente utilizadas no desenvolvimento de sistemas OLTP. Para o

estudo de caso foi utilizado o modelo de dados do sistema de Concurso Vestibular da UNIJUÍ .

## 8.2. Contribuições e limitações deste trabalho

Este trabalho apresenta contruibuições importantes aos profissionais envolvidos no processo de data warehousing dentre as quais destacamos: A apresenatação de três metodologias as quais fornecem uma visão bem detalhada das possibilidades e dificuldades inerentes ao processo de construção de um *data warehouse*. Adicionalmente, identificamos os pontos positivos e negativos de cada uma das metodologias apresentadas.

Com o objetivo de suprir as limitações das metodologias apresentadas, elaborou-se uma proposta de metodologia a qual representa um avanço em relação às metodologias apresentadas visto que contempla e detalha as várias fases do projeto e construção de um *data warehouse* valorizando a experiência adquirida pelos profissionais no desenvolvimento de sistemas transacionais com destaque para as fases de modelagem conceitual e projeto lógico.

Através do estudo de caso avaliou-se a metodologia proposta e constatou-se que ela pode ser utilizada integralmente como guia para o desenvolvimento de *data warehouse*. O grau de detalhamento e a distinção entre as fases do projeto são os principais pontos positivos a destacar.

Outra contribuição deste trabalho é a verificação da aplicabilidade da ferramenta de de projeto e desenvolvimento de *data warehouse* utilizada (IBM DB2 Warehouse Manager). Esta ferramenta mostrou-se muito apropriada facilitando o processo extração, transformação, limpeza e carga dos dados para o *data warehouse*.

Este trabalho foi inteiramente desenvolvido tendo em vista a abordagem relacional de banco de dados. Esta limitação é justificada pela atual predominância deste modelo no mercado. Em KLEIN (1999) encontramos uma análise da viabilidade e adequação do emprego da tecnologia de orientação a objeto (OO) em ambientes de *data warehouse* considerando a sua implementação em Sistemas de Gerenciamento de Bancos de Dados Relacional-Objeto (SGBDROs).

## 8.3. Sugestões para trabalhos futuros

O enfoque principal deste trabalho está relacionado á proposição de uma metodologia de desenvolvimento de *data warehouse* e avaliação da sua aplicabilidade através de um estudo de caso. Como extenção, enumeramos abaixo um conjunto de tópicos para trabalhos futuros.

- Identificação de uma ferramenta OLAP que ofereça as funcionalidades necessárias ao desenvolvimento de aplicações que exploram os recursos oferecidos pelo *data warehouse* projetado.
- Maior detalhamento do tópico relacionado à criação dos metadados.
- Avaliação da metodologia, através de estudo de caso, considerando diferentes arquiteturas de *data warehouse*. O estudo de caso desta dissertação avaliou a metodologia considerando o projeto de um *data warehouse* centralizado a nível global da empresa.
- Aplicabilidade da metodologia considerando uma equipe composta por profissionais sem experiência no desenvolvimento DW.
- A metodologia explora detalhadamente os níveis conceitual e lógico. Aspectos relacionados à implementação física do *data warehouse* (particionamento, indexação, materialização de visões) devem ser acrescidos a metodologia.
- Definir uma política de atualização do data warehouse.

# REFERÊNCIAS BIBLIOGRÁFICAS


BALLARD C.; HERREMAN D.; SCHAU D.;et al. ***Data Modeling Techniques for Data Warehousing***. IBM – ITSO redbooks, 1998.

BOEHNLEIN M.; ENDE A. ***Deriving Initial Data warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems***. Ulbrich-vom. Kansas City Mo USA, 1999.

BOHN, K. ***Converting Data for Warehouses***. DBMS Magazine. Junho, 1997.

CAMPOS, M. L.; ROCHA, A. V. ***Data warehouse***. Congresso SBBD, Outubro, 2001.

CAMPOS, M.L.M; BORGES, V.de J. A.S.**Diretrizes para a Modelagem Incremental de Data Marts**. Anais XVII SBBD, Outubro, 2002.

CHAUDHURI S.; DAYAL U. ***An Overview of Data Warehousing and OLAP Technology***. SIGMOD Record. Vol26 Num1, Setembro, 1997.

CODD, E.F., ***A Relational Model of Data for Large Shared Data Banks***. Communications of ACM. 13 (6), June 1970 377-387.

COME, Gilberto de. ***Os Metadados no ambiente de data warehouse***. disponível em: http://www.ead.fea.usp.br/Semead/4semead/Artigos/Mqi/Come.pdf. 1999.

FORSMAN, Sarah. ***OLAP Council white paper***. Disponível em: http://www.olapcouncil.org/research/whtpaply.htm, 2002;

GARCIA, S.S., et al. ***Ferramentas OLAP***. Disponível em: http://genesis.nce.ufrj.br/dataware/Tebdpos2001_3/index.htm. 2002.

GATZUI, Stella; STAUDT, Martin; VASSILIOU, Yannis, et al. ***Design and management of Data warehouses. Heidelberg***. Germany. 1999.

GATZUI, Stella; VAVOURAS, Athanasios. ***Data Warehousing: Concepts and Mechanisms***. Disponível em: http://www.svifsi.ch/revue/pages/issues/n991/a991Gatziu.pdf. 2002.

GOLFARELLI, Matteo; RIZZI, Steffano. ***A methodological framework for Data warehouse Design***. DOLAP 98 Washington, D.C., USA.

GOLFARELLI, Matteo; MAIO, Dario; RIZZI, Steffano. ***Conceptual Design of Data warehouses from E/R Schemas***. Proceedings of the Hawaii International Conference On System Sciences, Kona, Hawaii, January 1998.

GOLFARELLI, Matteo; RIZZI, Steffano. ***WAND: A CASE Tool for Data warehouse Design***. Disponível em: http://bias.csr.unibo.it/golfarelli/papers.htm. 2002

GREENFIELD, Larry. ***The Case For Data Warehousing***. Disponível em: http://www.dwinfocenter.org/casefor.html. 2002.

GUPTA, Vivek R., ***An Introduction to Data Warehousing***. Disponível em: http://www.system-services.com. 2002.

HAHN, Karl; SAPIA, Karsten; BLASCHKA, Markus. ***Automatically Generating OLAP schemata from Conceptual Graphical Models***. CIKM/DOLAP. Novembro, 2000.

HAYES, Scott; GUNNING, Philip. ***Tunning Up for OLTP and data warehousing***. DB2 Magazine.Vol 7 Num 3, 2002.

HERDEM, Olaf. ***A Design Methodology for Data warehouses***. Oldemburg Research and Development Institute for Computer Science Tools and Systems (OFFIS). Oldenburg, Germany.

HEUSER, Carlos Alberto. ***Projeto de banco de dados***. Porto Alegre: Editora Sagra Luzzatto, 2001.

IBM DB2 Universal Database. ***Business Intelligence Tutorial, IBM Corporation***. 2000.

INMON, William H. ***Como construir o data warehouse***. Tradução de Ana Maria Netto Guz. – Rio de Janeiro: Campus, 1997.

INMON, W.H.; WELCH, J.D.; GLASSEY, K.L. ***Gerenciando Data warehouse***. Tradução: Ana de Sá Woodward; São Paulo, Makron Books, 1999.

KELLY, Sean. ***The Data warehouse Toolkit***. Editora John Wiley & Sons Inc., New York, 1997.

KIMBALL, Ralph. ***Data warehousing in actiont***. Editora John Wiley & Sons Inc. New York, 1996.

KIMBALL, Ralph. ***A Dimensional Modeling Manifesto***. DBMS, http://www.dbmsmag.com/9708d15.html. Agosto, 1997.

KIMBALL, Ralph. ***Data warehouse Toolkit***. São Paulo: Editora Makron Books, 1998.

KLEIN, Lawrence Z. ***A Tecnologia Relacional-Objeto em um Ambiente de Data warehouse***. Dissertação de Mestrado IME, 1999.

KORTH, H. F.; SILBERSCHATZ, A. ***Sistemas de bancos de Dados***. Segunda Edição, Makron Books, 1993.

MACHADO, Felipe N. R. ***Projeto de Data warehouse: Uma Visão Multidimensional***. Editora Érica, 2000.

MADEIRA, Henrique. ***Tópicos avançados em bases de dados***. DEI-FCTUC, 2001.

MARCO, David. ***Revista Dados & negócios***. Edição Número 5, 2001.

McGUFF, Frank. ***Designing the perfect data warehouse***. Disponível em: http://www.gca.net/solutions/whitepapers/sybase/syb_dim_datamo_d.html

MOODY, Daniel L.; KORTINK, Mark A.R. ***From Enterprise Models to Dimensional Models: A Methodology for Data warehouse and Data mart Design***. (DMDW´2000).

MOORE, Arthur; WELLS, David. ***How To Do a Data warehouse Assessment (And Why)***. The Data Administration Newsletter. Disponível em: http://www.tdan.com/i013fe03.htm. 2002.

RODRIGUES, A. Q. et al. ***Ferramentas OLAP***. Diponível em:

http://genesis.nce.ufrj.br/dataware/Tebdpos2001_3/index.htm

SAMOS, José. et al. **Database Architecture for Data Warehousing**: **An Evolutionary Approach**. Disponível em: http://citeseer.nj.nec.com/samos98database.html, 1998.

PENDSE, Nigel. ***How to buy na OLAP product***. Disponível em:

http://www.olapreport.com/How_not_to_buy.htm. 2001.

PEREIRA, Walter Adel Leite. ***Trabalho Individual***. Disponível em:

http://www.inf.pucrs.br/~wpereira/. 1999.

OZSU, M. Tamer; VALDURIEZ, Patrick ***Principles of database systems***. Second Edition. Prentice Hall, 1999.

POE, V.; KLAUER, P.; BROBST, S. ***Building a Data warehouse for Decision Support***. Prentice Hall, Segunda Edição, 1998.

SAPIA, C.; BLASCHKA, M.; HOFLING, G.; DINTER, B. ***Extending the E/R Model for the Multidimensional Paradigm***. Proc of International Workshop on *Data warehouse* and Data Mining, November 1998.

SCHOUTEN, Han. ***Analysis and Design of warehouses***. Proc of International Workshop on Design and Management of *data warehouse*s, June 1999.

SINGH, HARRY S. ***Data warehouse, Conceitos, Tecnologias, Implementação e gerenciamento***. Tradução: Mônica Rosemberg. São Paulo, Makron Books, 2001.

TANNENBAUM, Adriane. ***Pensando metadados***. Dados & Negócios. Número 7, Primeiro trimestre de 2002.

TRYFONA, N.; BUSBORG, F.; CHRUSTUABSEN, J.G.B. ***starER: A Conceptual Model for Data warehouse Design***. ACM Second International Workshop on Data Warehousing and OLAP (DOLAP) 1999. Proceedings ACM Press.

UNIJUÍ , ***Base de dados, Série Cadernos da Avaliação Institucional*** . N.16, Segunda Edição, 1998.

VASSILIADIS, Panos. ***Gulliver in the land of data warehousing: practical experiences and observations of a researcher***. Athens, Greece. June 2000.