

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2004.2

MODELAGEM INCREMENTAL DE UM AMBIENTE DE DATA WAREHOUSING: UMA ABORDAGEM DA ARQUITETURA BOTTOM-UP

TRABALHO DE GRADUAÇÃO
EM
BANCO DE DADOS

Aluno: José de Anchieta C. Caraciolo Netto, jaccn@cin.ufpe.br.
Orientador: Fernando da Fonseca de Souza, fdfd@cin.ufpe.br.

Recife, 11 de Março de 2005.

AGRADECIMENTOS

Não poderia de deixar de prestar meus profundos agradecimentos a Fernando Fonseca, meu orientador, ser humano fantástico, cujas sugestões me incentivaram a abrir os olhos para uma área que conquistou meu interesse e me forneceu uma sólida base de conhecimentos que, inevitavelmente, abrirá novos horizontes em minha vida.

Agradeço a minha noiva, Raissa, por toda sua compreensão, ternura e carinho, ao lado de quem encontrarei a felicidade e paz por toda a vida.

Agradeço aos meus pais, que me presentearam com a minha vida, minha educação, e ainda que eu agradecesse eternamente, jamais seria suficiente pela compreensão que eles tiveram comigo.

Agradeço também aos meus amigos, aqueles que me ajudaram a levantar em momentos difíceis e, principalmente, ao meu eterno amigo Mauro Quaresma, cujas lições de vida me acompanharão durante todos os meus dias.

E, finalmente, um agradecimento especial as minhas irmãs Geórgia e Juliana e a todos que torceram por mim, que me acompanharam nesta longa jornada e agora dividem comigo essa felicidade. Obrigado.

RESUMO

O aumento da demanda e a valorização das informações estratégicas em sistemas de apoio à decisão têm sido uma das grandes razões para investimentos em soluções informatizadas, relacionadas principalmente à tecnologia de Data Warehousing. Porém, as constantes mudanças dos cenários no processo de decisão requisitam a utilização de uma metodologia de construção de Data Warehousing que favoreça a flexibilidade focada num processo evolutivo contínuo, viabilizando o acompanhamento das mudanças das necessidades dos tomadores de decisões. Seguindo esta metodologia, este trabalho propõe uma arquitetura que permita grande flexibilidade e rapidez no desenvolvimento, além das vantagens trazidas para os analistas de negócios, como resultados mais rápidos e adaptados às suas necessidades. Esta arquitetura, aplicada com um conjunto de técnicas sugeridas neste trabalho, possibilitará realizar a modelagem de dados em um ambiente de Data Warehouse de forma incremental, permitindo o constante acompanhamento do usuário final.

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO.....	1
1.1 JUSTIFICATIVA.....	2
1.2 OBJETIVOS	3
1.3 ESTRUTURA DO TRABALHO.....	4
CAPÍTULO 2 – DATA WAREHOUSING	5
2.1 DEFINIÇÃO	5
2.2 PRINCIPAIS CARACTERÍSTICAS DO DATA WAREHOUSE.....	6
2.3 MODELO DE FUNCIONAMENTO	10
2.4 FASES TÍPICAS NO DATA WAREHOUSING	12
2.4.1. Planejamento	12
2.4.2 Modelagem Lógica e Física	16
2.4.3 Preparação e Desenvolvimento de Ferramentas de Extração	24
2.4.4 Desenvolvimento de Ferramentas de Aplicações Cliente	27
2.4.5 Metadados	29
2.5 METODOLOGIAS DE DESENVOLVIMENTO	30
CAPÍTULO 3 – ABORDAGEM PROPOSTA PARA O DESENVOLVIMENTO INCREMENTAL DE UM DATA WAREHOUSE	35
3.1 ARQUITETURAS	35
3.1.1 Tipos de Arquiteturas	37
3.2 A ARQUITETURA PROPOSTA	41
3.3 METODOLOGIA DE DESENVOLVIMENTO ADOTADA.....	42
CAPÍTULO 4 – APLICAÇÃO: SISTEMA DE APOIO A DECISÃO PARA UMA FÁBRICA NACIONAL DE AUTOMÓVEIS	46
4.1 DATA MART MONTADORA	47
4.2 DATA MART CONCESSIONÁRIA	54
4.3 ANÁLISE DOS RESULTADOS.....	61
CAPÍTULO 5 – CONCLUSÃO E TRABALHOS FUTUROS	63
5.1 TRABALHOS FUTUROS.....	64
REFERÊNCIAS BIBLIOGRÁFICAS.....	66
ANEXO A.....	69
ANEXO B	70

LISTA DE FIGURAS

Figura 1: Arquitetura Usual do Ambiente de Data Warehouse.....	2
Figura 2: DW – Dados para Informação.....	6
Figura 3: Definição do Data Warehouse.....	7
Figura 4: Orientação por Assunto do Data Warehouse.....	8
Figura 5: Processamento básico de um Data Warehouse.....	11
Figura 6: Fases típicas no desenvolvimento de um DW.....	12
Figura 7: Etapas do planejamento.....	12
Figura 8: Modelagem Lógica e Física.....	16
Figura 9: O “Cubo”.....	17
Figura 10: Representação do Esquema Estrela.....	19
Figura 11: Desenvolvimento das Ferramentas de Extração.....	24
Figura 12: Processo de Extração.....	25
Figura 13: Desenvolvimento da área de apresentação.....	27
Figura 14: Desenvolvimento de Metadados.....	29
Figura 15: Ciclo de Vida Tradicional de Desenvolvimento.....	31
Figura 16: Etapas do Desenvolvimento com Data Marts Incrementais.....	33
Figura 17: Arquitetura Top-Down.....	39
Figura 18: Arquitetura Bottom-Up.....	39
Figura 19: Diretrizes para a Modelagem de um ADW Incremental.....	45
Figura 21: DER Resultante da Fase A e B.I.a.....	49
Figura 22: DER Resultante da Fase de Limpeza.....	51
Figura 23: Data Mart Montadora.....	54
Figura 24: DER Concessionária.....	55
Figura 25: DER Resultante da fase B.I.a.....	56
Figura 26: DER Resultante da Fase de Limpeza.....	57
Figura 27: Data Mart Concessionária.....	60
Figura 28: Data Warehouse Global.....	61
Figura 29: Valores distribuídos no tempo e agrupados por concessionárias e estados.....	70
Figura 30: Drill-down nas concessionárias.....	71

Figura 31: Drill-down nos estados.....	72
Figura 32: Representação gráfica dos valores por concessionária e por Estado.....	73

LISTA DE QUADROS

Quadro 1: Funcionalidades Necessárias aos Processos de ETC	9
Quadro 2: Ciclo de desenvolvimento orientado ao projeto	31
Quadro 3: Metodologia Top-down.....	32
Quadro 4: Desenvolvimento com Data Marts Incrementais	33
Quadro 5: Arquitetura Top-Down	38
Quadro 6: Arquitetura Bottom-Up.....	40
Quadro 7: Regras de definição das Categorias.....	50
Quadro 8: Regras para a Substituição de Códigos e Abreviaturas	81
Quadro 9: Regras para a Substituição de Códigos e Abreviaturas	58
Quadro 10: Papéis em um Projeto de Data Warehouse.....	69

CAPÍTULO 1

INTRODUÇÃO

O desenvolvimento da tecnologia de informação, aliado à globalização e ao aumento da competitividade nos mais variados setores, está facilitando a integração entre o mercado produtor e consumidor. A cada negócio realizado, ou seja, a todo o momento, uma grande quantidade de dados é gerada e armazenada, passando a ser um recurso importante da empresa.

O grande problema está em como tratar esse grande volume de dados, geralmente espalhados por diversos sistemas de informação, de forma a obter alguma vantagem competitiva através da informação certa no momento certo.

O Data Warehouse surgiu nos anos 90 [BER1997] como uma maneira flexível e eficiente de tratar esses grandes volumes de dados e obter informações que auxiliem no processo para tomada de decisão, mas sua implementação não é tão simples. Projetos dessa natureza demandam uma grande quantidade de tempo e dinheiro e portanto precisam de uma atenção especial.. A abordagem de Data Warehouse permitiu aperfeiçoar a gerência, o controle e o acesso aos dados, e tornou este ambiente conhecido por Ambiente de Data Warehouse (ADW) [IMN1997]. A arquitetura usual deste ambiente pode ser representada pela figura 1.

Os números de uma pesquisa do Instituto Standish mostram que, mesmo utilizando as técnicas de Data Warehousing, muitos sistemas ainda são mal sucedidos. De acordo com estes estudos, 31% dos projetos são cancelados, enquanto que o percentual de atraso na entrega dos projetos e orçamentos estourados chega a 53% [DAT2001]. Esses resultados, em sua maioria, são causados pelo desenvolvimento baseado em uma seqüência rígida de tarefas, onde pouca ênfase sobre flexibilidade é dada na concepção da arquitetura do sistema.

Este trabalho tem como objetivo sugerir uma arquitetura que permita grande flexibilidade e rapidez no desenvolvimento, além de indicar um conjunto de regras que permita realizar a modelagem de dados de um ambiente de Data Warehouse de forma incremental. Assim, os primeiros resultados são apresentados rapidamente ao usuário, sem perder a visão integrada do negócio, além de permitir uma rápida absorção de novos requisitos, dada a flexibilidade proporcionada pela arquitetura.

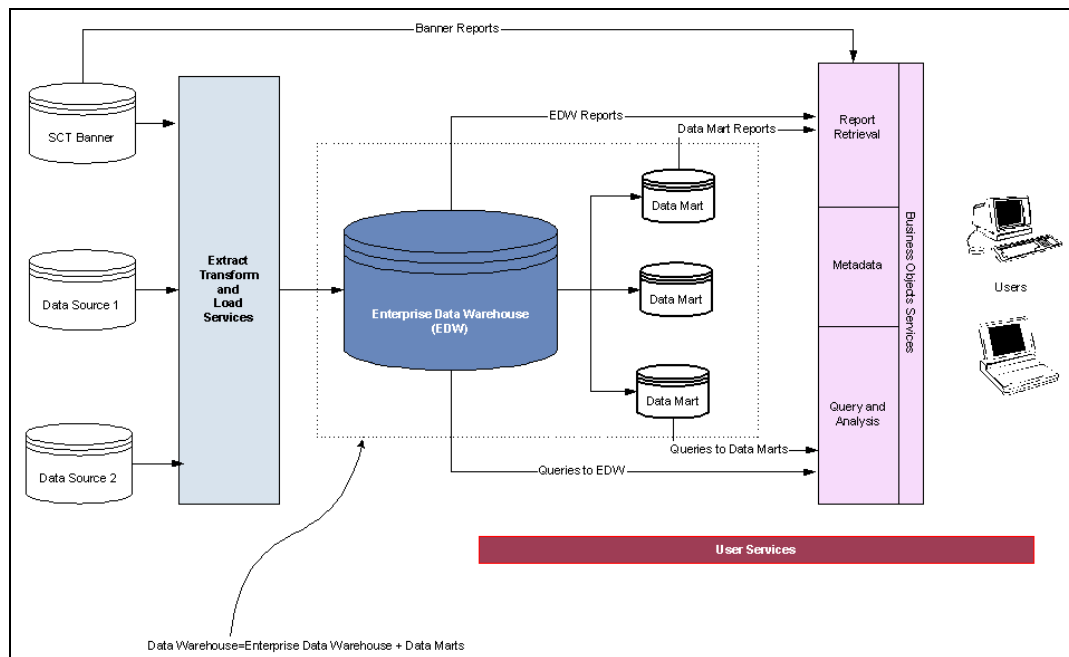


Figura 1: Arquitetura Usual do Ambiente de Data Warehouse
Fonte: UNIVERSITY OF ILLINOIS, 2005 [UNI2005]

1.1 JUSTIFICATIVA

A utilização das metodologias e arquiteturas usuais mostrou através de vários insucessos em projetos, ser inadequada para a implementação de grandes sistemas de informações.

A modelagem padrão normalmente utilizada primeiramente desenvolve o modelo global do Data Warehouse para depois gerar os data marts. Esta abordagem é conhecida como *Top-down* não apresenta bons resultados, visto que seu processo de modelagem é bastante custoso, pouco prático e sua implementação requer muito tempo.

Grande parte das empresas prefere um tipo de abordagem que possibilite uma rápida apresentação de resultados, de maneira incremental. Esta forma de abordagem é satisfeita com a arquitetura *Bottom-up*.

Após a escolha correta de uma arquitetura, é necessário o uso de um conjunto de técnicas que favoreça uma metodologia de desenvolvimento incremental nos projetos tecnológicos de sistemas de informações, de forma que se possa apresentar resultados de forma mais rápida e acompanhar as mudanças de requisitos dos usuários finais. Isto é possível devido à independência dos módulos componentes da arquitetura, justificando assim, os investimentos nos projetos de tecnologia da informação.

1.2 OBJETIVOS

O objetivo deste trabalho de graduação é sugerir, baseado no material bibliográfico utilizado, uma arquitetura para organização do Data Warehouse, de forma a torná-lo flexível a mudanças nos requisitos de projeto e indicar um conjunto de técnicas que permita realizar, a partir dos modelos de dados existentes no ambiente operacional, a modelagem de dados de um ambiente de Data Warehouse de forma incremental. A modelagem do ambiente será realizada a partir da criação de data marts, por um processo de derivação dos modelos de dados do ambiente operacional, e sua posterior integração ao Data Warehouse, caracterizando assim uma arquitetura *Bottom-up*.

Todas as etapas da metodologia de desenvolvimento incremental serão demonstradas, em quatro fases distintas, para tornar a seqüência das técnicas mais clara [SOA1998]. As quatro fases podem ser divididas da seguinte forma:

- Análise dos modelos de dados existentes no ambiente operacional.
- Transformação do modelo de dados, até então orientado a processo, em um modelo de dados mais próximo ao negócio.
- Derivação de um ou mais modelos dimensionais para o data mart, a partir do modelo de dados resultante da fase anterior.

- Integração do modelo de dados do data mart ao modelo de dados do Data Warehouse, finalizando desta forma a modelagem do Ambiente de Data Warehouse.

A arquitetura sugerida possui o processo de carga semelhante aos padrões normais do ambiente de Data Warehouse (*Top-down*). Ou seja, os dados são extraídos das fontes, sistemas operacionais e fontes externas e carregados para o Data Warehouse, após um processo de limpeza e transformação de dados, para depois serem disponibilizados para os data marts. Porém, as principais características da arquitetura Bottom-up são mantidas de forma que, mesmo utilizando este tipo de alimentação de dados, a abordagem não deva ser considerada como intermediária.

1.3 ESTRUTURA DO TRABALHO

Além deste capítulo introdutório contendo a contextualização, justificativa e objetivos, fazem parte deste trabalho ainda os seguintes capítulos:

Capítulo 2 – Principais elementos, fases da construção e aspectos de implementação da técnica de Data Warehousing.

Capítulo 3 – Principais características das arquiteturas empregadas em sistemas de informações baseados em Data Warehouse. Será analisada neste capítulo a arquitetura proposta, suas diferenças em relação às outras arquiteturas e benefícios da sua utilização, além da proposta de uma metodologia de desenvolvimento baseada em data marts incrementais.

Capítulo 4 – Aplicação das técnicas propostas em um estudo de caso de uso em um ambiente de uma grande empresa automobilística nacional. Este estudo consiste no desenvolvimento do ambiente de Data Warehouse (ADW) da empresa, através da modelagem de dois data marts: O data mart Montadora e o data mart Concessionária e a integração incremental dos mesmos ao Data Warehouse global.

Capítulo 5 – Conclusão geral do trabalho, com sua contribuição e sugestões de trabalhos futuros.

CAPÍTULO 2

DATA WAREHOUSING

O conceito de data warehousing (desenvolvimento de sistemas de informações baseados em Data Warehouse), apesar de recente, é baseado em idéias que vinham sendo aplicadas em vários sistemas de informações há muitos anos [INM1997]. A técnica foi sendo aperfeiçoada com o surgimento e evolução de várias tecnologias e metodologias, que facilitam a implementação dos sistemas de informações.

A arquitetura proposta foi elaborada para a distribuição dos componentes de um Data Warehouse, e pode ser aplicada em conjunto com a metodologia de desenvolvimento baseada em data marts incrementais. Tem por objetivo orientar a implementação dos vários componentes tecnológicos existentes em um sistema de informações.

Neste capítulo aborda-se a técnica data warehousing, seus principais elementos, fases da construção e aspectos de implementação. Dentre estes aspectos, um maior nível de detalhe será dado às metodologias empregadas para o desenvolvimento do ambiente de Data Warehouse.

2.1 DEFINIÇÃO

Data Warehousing é uma técnica de desenvolvimento do componente tecnológico de sistemas de informações onde a preparação dos dados e do ambiente é baseada em um Data Warehouse. Segundo Singh [SIN2001], Data Warehouse é uma tecnologia de gestão e análise de dados, constituindo “um ambiente de suporte a decisão que alavanca dados armazenados em diferentes fontes e os organiza e entrega aos tomadores de decisões da empresa, independente de plataforma que utilizam ou de seu nível de qualificação técnica”. O propósito do Data Warehouse (DW) não é suportar as operações ou transações da organização, mas centralizar os

dados extraídos de aplicações operacionais e bancos de dados. Em geral, um DW requer a consolidação de outros recursos de dados além dos armazenados em banco de dados relacionais, incluindo informações provenientes de planilhas eletrônicas, documentos textuais e outros. O DW não é construído para suportar o processo funcional ou operacional da empresa, ou seja, não é o fim, mas é o meio para facilitar o uso da informação, como mostra a Figura 2 [INM1997, KIM1998 e SIN2001].

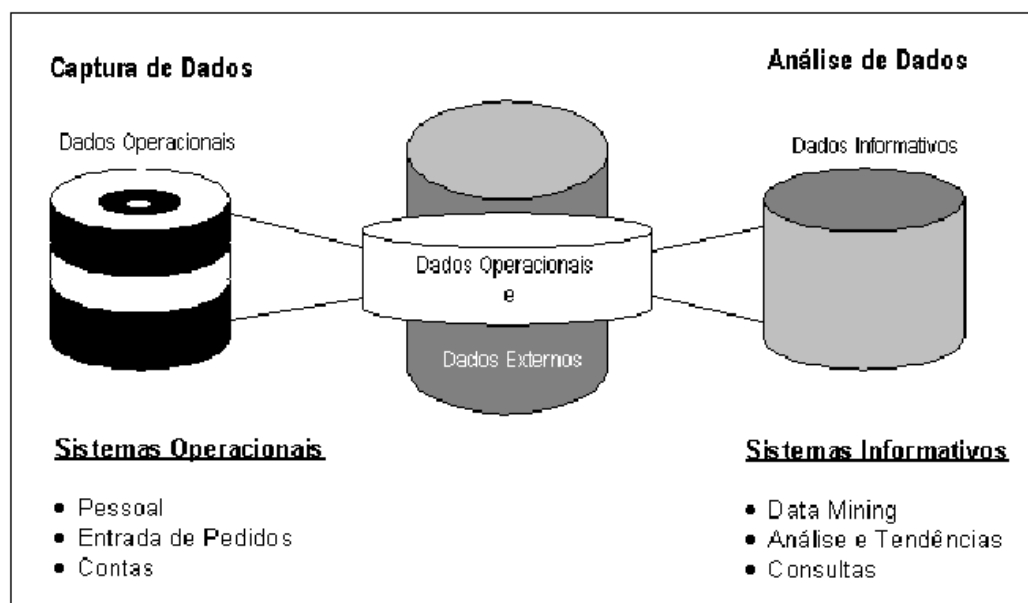


Figura 2: DW – Dados para Informação
Fonte: SINGH, 2001 [SIN2001]

2.2 PRINCIPAIS CARACTERÍSTICAS DO DATA WAREHOUSE

Inmon [INM1997], define o Data Warehouse como “um conjunto de dados com as seguintes características: orientados por assunto ou negócio, integrados, variáveis com o tempo e não-voláteis, e que fornecem suporte ao processo de tomada de decisão do negócio”, como visto na figura 3. A seguir serão detalhadas essas características.

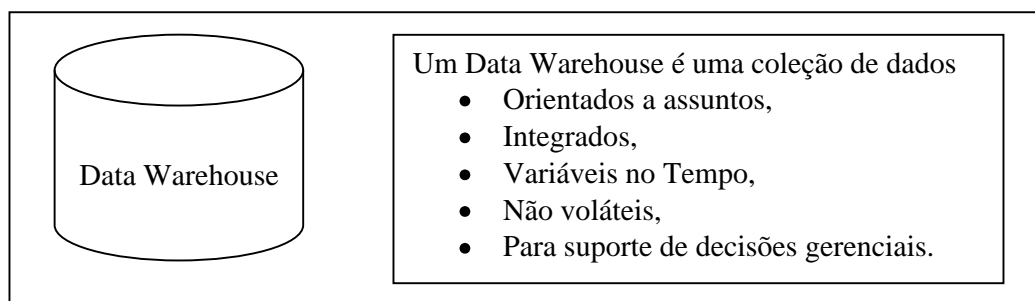


Figura 3: Definição do Data Warehouse
Fonte: Adaptado de Inmon, 1997 [INM1997]

- Orientado a assunto

Os dados operacionais ou funcionais são organizados em torno das aplicações da empresa, executam e registram as transações rotineiras necessárias para conduzir o negócio. O Data Warehouse baseia-se nos principais assuntos ou negócios da organização que tenham sido definidos no modelo de dados [INM1997 e LAU1999].

Os dados orientados a aplicações possuem detalhes que satisfazem os requisitos imediatistas do processamento funcional. Esses detalhes podem ser irrelevantes ao analista de SAD (suporte a decisão). O Data Warehouse não inclui dados que não serão usados para processamento SAD [INM1997].

Inmon [INM1997], exemplifica o caso de uma companhia de seguros, onde os dados são organizados em torno de aplicações (Sistemas de Informações de Processamento de Informações – SPT) que podem ser automóvel, saúde, vida e perdas. O Data Warehouse da companhia é baseado em assuntos ou negócios da empresa que podem ser cliente, apólice, prêmio indenização, como mostra a Figura 4.

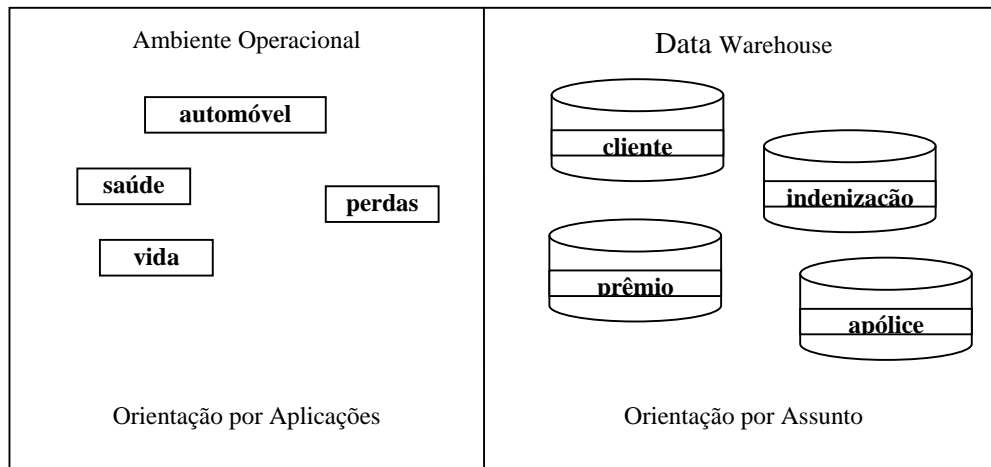


Figura 4: Orientação por Assunto do Data Warehouse

Fonte: Adaptado de Inmon, 1997 [INM1997]

- Integrado

A característica mais importante do ambiente de Data Warehouse, é a sua integração interna. A integração de dados ocorre quando os dados são passados do ambiente operacional baseado em aplicações para o ambiente de DW [INM1997].

As aplicações não apresentam coerência, surgem ao longo dos anos de sob formas diferentes. A integração dos dados de diversas aplicações é feita de diferentes formas na consistência de nomes, na consistência de variáveis de medidas, na consistência da codificação das estruturas, na consistência dos atributos físicos dos dados e assim por diante. Outro aspecto importante é que a codificação para o Data Warehouse deve ser feita de forma independente da aplicação de origem [INM1997].

Os processos de extração, transformação e carga (ETC), executam a grande e complexa tarefa da passagem dos dados do ambiente operacional herdado para o ambiente Data Warehouse. O quadro 1 apresenta algumas funcionalidades necessárias aos processos de ETC .

Quadro 1: Funcionalidades Necessárias aos Processos de ETC

Funcionalidades necessárias
A extração de dados do ambiente operacional para o ambiente de DW demanda mudança de tecnologia, por exemplo, SGBD de DW.
A seleção e validação dos dados do ambiente operacional.
As chaves de entrada operacionais geralmente precisam ser reestruturadas antes de serem gravadas. Em casos simples, um elemento de tempo é acrescentado à estrutura da chave.
Os dados são reformatados
Trabalhar com várias fontes de dados. A lógica deve ser esclarecida para que a fonte de dados apropriada contribua com seus dados segundo o conjunto correto de condições.
Intercalar vários arquivos de entrada.
Trabalhar com vários resultados de diferentes níveis de resumos produzidos pelo mesmo programa de carga.
A eficiência na escolha dos dados de entrada para extração.
Freqüentemente é necessário resumir dados.
A alteração de nomes de elementos de dados durante a passagem do ambiente operacional para o ambiente de <i>Data Warehouse</i> deve ser registrada.
Os registros de entrada com formatos não-padronizados.
A conversão precisa ser feita.
Talvez o pior de tudo: Os relacionamentos entre os dados, que embutidos nos antigos programas herdados precisam ser compreendidos. Geralmente esses relacionamentos são misteriosos, difíceis de alterar e não-documentados.
Processamento com volumes massivos de entrada de dados. Implementação de cargas em paralelo ou leitura em paralelo.
O projeto do DW deve adequar-se ao modelo de dados corporativos. A entrada para o DW ajusta-se às especificações de uma aplicação que foi escrita há muito tempo.

Fonte: INMON, 1997 [INM1997]

- Variação com o tempo

Segundo Singh [SIN2001] “variação com o tempo significa que os dados estão associados a um ponto no tempo (ou seja, semestre, ano fiscal e período de pagamento)”.

Para Inmon [INM1997], todos os dados no Data Warehouse são precisos em algum instante no tempo. No ambiente operacional, os dados estão corretos como no momento do acesso. Em virtude dos dados no DW serem corretos como em algum momento no tempo, “não exatamente agora”, é dito que estes dados “variam com o tempo” [INM1997].

No nível operacional atuam diversos Sistemas de Informações de Processamento de Transações (SPT), também chamados de Processamento

Transacional On-Line (*On- Line Transaction Processing – OLTP*). Já no Data Warehouse o armazenamento é uma seqüência de tempo explícita, onde são transferidos instantâneos estáticos (*static snapshots*) do OLTP para o DW como uma seqüência de camada de dados, muito semelhante às formações geológicas. O Data Warehouse é uma seqüência de tempos, e como os geólogos, pode-se escavar as camadas para saber como eram os negócios do passado. Essa técnica de DW é chamada de mudança gradual nos eixos temporais (*slowly changing dimensions*), é considerada a principal técnica para representar o passado, onde as duas inconsistências temporais nos bancos de dados OLTP são resolvidas. A primeira, o Data Warehouse não se modifica durante o dia quando os usuários estão fazendo as suas consultas. A segunda, armazenando as informações cuidadosamente em cada instantâneo no DW, pode-se representar todos os pontos de tempo anteriores corretamente [KIM1998].

- Não-volátil

No ambiente de Data Warehouse (ADW), não-volátil significa que os dados não são alterados depois de inseridos. Os bancos de dados operacionais/OLTP mudam todas as vezes que uma transação é processada. No ADW, alterações instantâneas não são permitidas [SIN2001, TAN1998 e KIM1998].

Inmon [INM1997], afirma que no ADW os dados são inseridos e acessados em grandes quantidades, mas a atualização dos dados geralmente não ocorre no ADW. No ambiente operacional, os dados são regularmente acessados um registro por vez e podem ser atualizados.

2.3 MODELO DE FUNCIONAMENTO

Para responder os questionamentos dos usuários em um Data Warehouse, parte-se do pressuposto de que o acesso aos dados seja efetuado sobre uma (ou várias) bases de dados consolidadas.

Para tanto, o esquema de funcionamento de um de Data Warehouse pode ser dividido em processos básicos, sendo estes a extração de dados dos sistemas operacionais, o armazenamento dos dados e a apresentação de informações, conforme ilustrado na figura 5.

Por extração de dados, entende-se a concepção ou aquisição e parametrização das ferramentas que realizarão as tarefas de coleta, limpeza, transformação e migração dos dados operacionais ao Data Warehouse. A realização de todas as tarefas desta fase constitui um dos processos mais lentos e delicados no Data Warehousing [KIM1998 e INM1997].

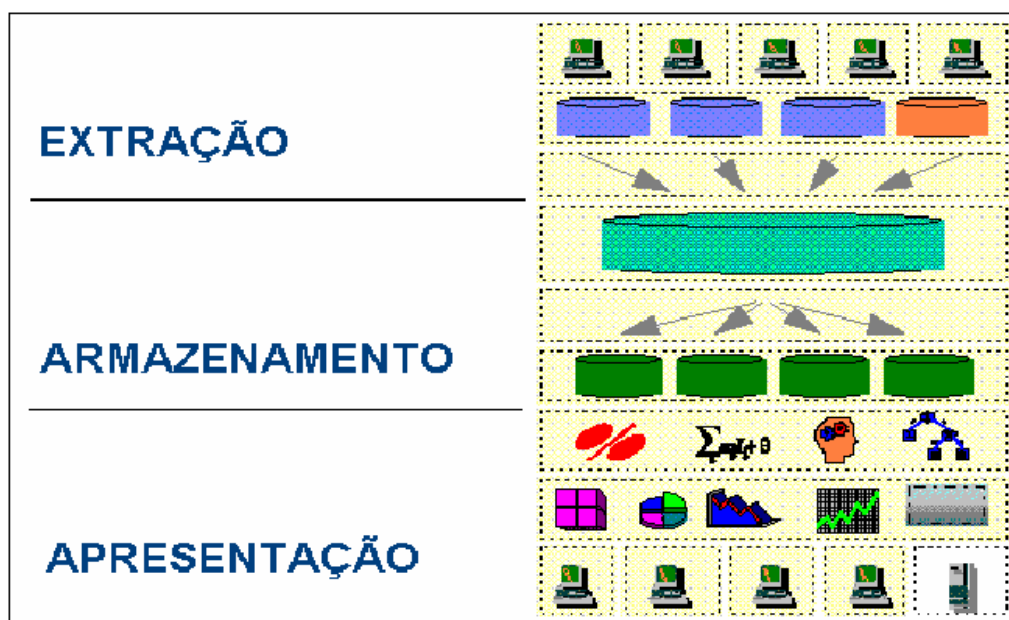


Figura 5: Processamento básico de um Data Warehouse

Fonte: Adaptado de Armstrong, 2005 [ARM2005]

Por armazenamento dos dados, entende-se a concepção do repositório das informações, que é o núcleo do ambiente do Data Warehouse. Neste estarão representados todos os dados extraídos dos sistemas operacionais, necessários para o processo de tomada de decisão [INM1997 e KIM1998].

Por apresentação de informações entende-se a concepção ou aquisição e parametrização das aplicações clientes e do servidor que atenderá às requisições de

dados junto ao Data Warehouse e disponibilizará as informações resultantes [TAN1998 e COR1998].

2.4 FASES TÍPICAS NO DATA WAREHOUSING

O desenvolvimento do Data Warehouse pode ser decomposto em processos distintos conforme figura 6.

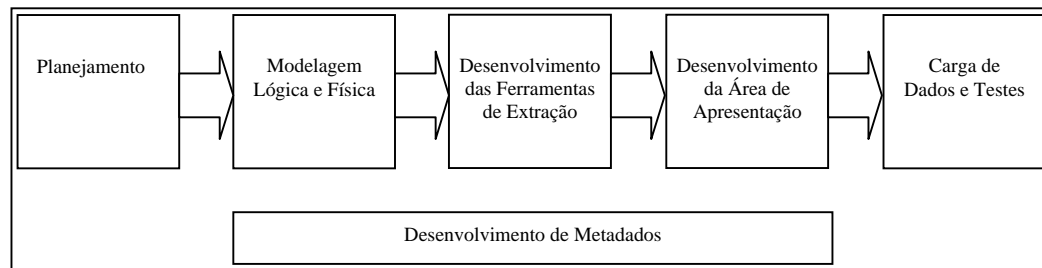


Figura 6: Fases típicas no desenvolvimento de um DW

Fonte: Adaptado de GIOVINAZZO, 2000 e DEBEVOISE, 1999 [GIO2000 e DEB1999]

2.4.1. Planejamento

Como o próprio nome propõe, nesta fase é planejada a construção do sistema. Constitui uma das fases mais importantes, pois qualquer falha na delimitação de escopo, identificação de necessidades ou erro na especificação dos recursos pode resultar na inviabilização total do projeto. As etapas constituintes desta fase são ilustradas na figura 7 e em seguida cada qual será abordada [KIM1998].

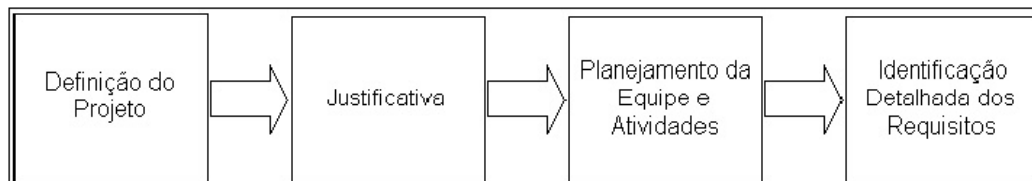


Figura 7: Etapas do planejamento

Fonte: Adaptado de KIMBALL, 1998 [KIM1998]

a. Definição do Projeto

Nesta importante fase do planejamento, é identificada a existência e a origem de demanda. Logo, será realizado um levantamento da empresa, onde serão identificados os potenciais usuários e suas necessidades.

Segundo Kimball [KIM1998], para o sucesso de um projeto de Data Warehouse é necessário que a empresa esteja realmente engajada no projeto, devido à necessidade de acompanhar os concorrentes ou estratégia de crescimento. Outro fator importante é a parceria entre comunidade de usuários e profissionais de informática e a observação da cultura de utilização de informação na empresa [CHU1998].

Em caso de identificação de pouca necessidade, falta de patrocinadores¹ influentes e motivados ou falta de cultura do uso de informação, recomenda-se o adiamento do projeto, antes de obter-se um fracasso no meio do projeto.

Nesta etapa deve-se focar brevemente os pontos estratégicos do negócio e como são acompanhados, bem como identificar os processos centrais da organização e quais são mais desejáveis para serem abordados no projeto.

b. Justificativa

Definido o escopo, o passo seguinte é construir a justificativa do projeto. Nesta justificativa serão apresentados os custos e benefícios estimados.

Entre os custos, deverão ser considerados [KIM1998 e INM1997]:

- aquisição de *hardware* e licenças de software;
- despesas com manutenção do *hardware* e software;
- estimativa inicial de recursos humanos internos e externos;
- preparação do corpo técnico e treinamento de usuários;
- despesas com equipe de suporte;
- despesas com upgrades no *hardware*.

1. Usuários com respaldo político, responsáveis pela defesa do projeto e obtenção de recursos. Essenciais para um projeto de Data Warehouse [KIM1998].

Para determinar os benefícios, é necessário investigar os ganhos no processo de tomada de decisão da corporação. Deve-se priorizar a receita e o lucro obtidos nas oportunidades geradas.

Uma forma proposta por Kimball [KIM1998] é a de quantificar em dinheiro os retornos esperados, realizando questionamentos com os futuros usuários em relação às informações que serão obtidas pelo sistema, tais como:

- o que aconteceria se você tivesse...?
- você é capaz de prever...?
- quanto custa para a empresa não saber...?

Com as respostas obtidas neste primeiro questionamento, deve-se continuar a investigação indagando até obter-se valores monetários de ganhos e cortes de despesas.

Após a estimativa dos custos e dos benefícios, deverá ser calculado o retorno do investimento (*Return of Investment* – ROI). Deve-se salientar não só os retornos diretos, mas os retornos que poderão ser obtidos futuramente nos processos decisórios. Kimball [KIM1998] sugere que se deva apresentar como um “custo de oportunidade”, ou seja, se o benefício estimado com a implantação do sistema é de R\$ 100.000,00 mensais, então a sua não disponibilização custará R\$ 100.000,00 mensais para a empresa.

c. Planejamento da Equipe e Atividades

Tendo a aprovação para iniciar o projeto, deve-se definir a equipe e o plano de atividades do projeto.

A equipe será definida tendo em vista as atividades ou papéis constituintes de um projeto de Data Warehouse, levando-se em consideração que os membros da equipe assumirão mais de uma atividade/papel durante o projeto.

Cabe ao gerente do projeto definir os integrantes do projeto, observando os papéis existentes, recursos e tempo, além de esboçar o planejamento do projeto. Os

principais papéis relacionados por Kimball [KIM1998] são apresentados no anexo A de acordo com a ordem em que entram em cena no decorrer do projeto.

No planejamento do projeto serão detalhadas as tarefas, prazos, status e responsáveis. Na especificação das tarefas deve-se atentar para o maior desmembramento possível em sub-tarefas, a fim de ter um maior controle e cautela na estimativa de tempo [KIM1998 e INM1997].

d. Identificação Detalhada dos Requisitos

Esta consiste numa tarefa crucial para o sucesso do projeto. Na identificação dos requisitos são levantadas de forma detalhada as necessidades dos usuários.

A atividade é coordenada de preferência pelo analista da aplicação cliente. Frequentemente o gerente do projeto, modelador de dados, e o desenvolvedor da aplicação cliente auxiliam os procedimentos de identificação dos requisitos.

Existem muitas maneiras de se realizar a coleta de requisitos, mas de forma geral, estas seguem basicamente dois princípios: dirigida pelos dados ou dirigida pelos requisitos [CHU1998].

Quando dirigida pelos requisitos, serão levantadas as funções e as necessidades dos usuários. A coleta de requisitos se dá através de entrevistas com executivos, gerentes intermediários e funcionários. Além das entrevistas, investigações mais minuciosas são realizadas, através da análise dos sistemas existentes, questionários, sessões JAD (*Joint Application Design*) ou *brainstorming* e análise do plano estratégico de negócios [INM1997 e CHU1998].

Quando dirigido pelos dados, inicialmente serão levantados os dados existentes nos sistemas da organização, para verificar-se o que é possível oferecer. Neste caso corre-se o risco de não atender as reais necessidades dos usuários, mas dificilmente haverá problemas quanto a especificações que não possam ser implementadas por falta de dados.

Os resultados da definição dos requisitos são documentados e ao final da fase da coleta descrevem-se os requisitos do sistema em um documento que será revisado pelos futuros usuários.

2.4.2 Modelagem Lógica e Física

A compreensão dos dados é um dos maiores problemas no desenvolvimento do Data Warehouse. A construção do modelo de dados é fundamental para o desenvolvimento, ajudando a compreender as regras de negócio e a organização dos dados para o melhor tempo de resposta [KIM1998]. A figura 8 representa a fase da modelagem lógica e física dentro das fases típicas do desenvolvimento do Data Warehouse.

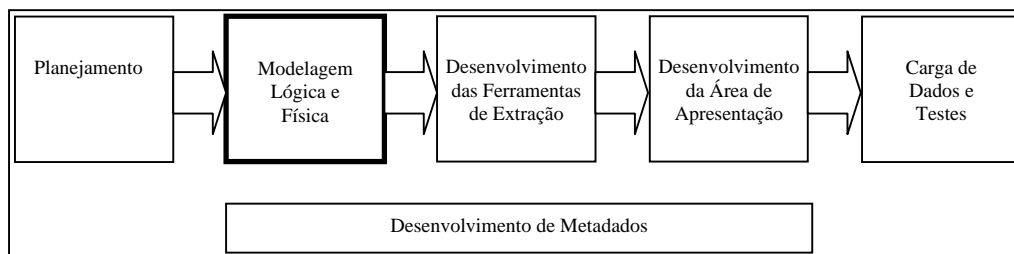


Figura 8: Modelagem Lógica e Física

Segundo Singh [SIN2001], o modelo de dados do DW representa os requisitos de informações integradas, os requisitos de análise e o suporte à decisão de toda a organização.

A modelagem de dados para DW é completamente diferente da modelagem utilizada nos sistemas operacionais (OLTP) [MAC2000]. Nos ambientes operacionais, o modelo utilizado é o Modelo Entidade/Relacionamento (MER) que utiliza uma técnica que busca remover qualquer redundância de dados, qualquer transação que atualize os dados será efetuada em apenas um ponto do banco de dados [KIM1998]. No ambiente de DW, é utilizado o modelo dimensional que utiliza uma técnica que suporta o ambiente para análise multidimensional dos dados [MAC2000].

Os diagramas de entidade/relacionamento são difíceis de visualizar e memorizar, são simétricos, isto significa que a aparência do diagrama pode ser modificada arbitrariamente, mas sem mudar o significado lógico do diagrama. As tabelas parecem iguais, são representadas da mesma forma ficando difícil identificar qual delas é a maior ou a mais importante, e quais tabelas contêm valores numéricos do negócio [KIM1998]. Kimball [KIM1998], afirma que “modelos de entidade/relacionamento são um desastre para consultas porque não podem ser entendidos pelo usuário nem navegados de forma útil pelo software de SGBD. Os modelos entidade/relacionamento não podem ser utilizados como base para Data Warehouse”.

- Modelo Dimensional

O modelo dimensional é geralmente visualizado através do desenho de um cubo. Um cubo pode conter três ou mais dimensões cada uma representando um atributo diferente conforme apresenta a Figura 9.

Como exemplo, tomam-se as dimensões de análise de dados de vendas *Produtos*, *Região* e *Trimestre*, e pode-se localizar um determinado fato, como o número de produtos da linha branca na região Nordeste. Desta forma o usuário pode “girar” o cubo para obter novos fatos a partir das dimensões.

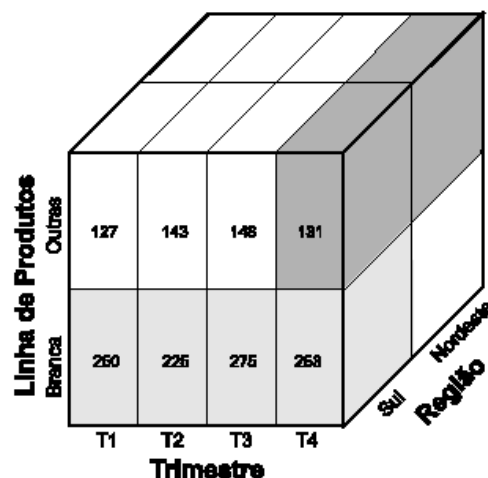


Figura 9: O “Cubo”

Os modelos dimensionais são construídos em torno de fatos numéricos de medida em um instante específico de tempo ou de medida acumulada sobre um período de tempo. Esses fatos são as medições numéricas do negócio, que podem ser a venda de um produto no varejo, o preço de uma ação em um determinado momento de tempo ou a alteração do salário por promoção. Os fatos medidos são coletados dos sistemas de computador de produção e colocados em tabelas denominadas, apropriadamente, tabelas de fatos [KIM2000].

Os elementos que participam de um fato (medida do negócio) são as dimensões [MAC2000]. As dimensões são descrições textuais das dimensões do negócio no momento que o fato é gerado, em síntese, são os atributos de um fato [KIM1888].

As descrições textuais de cada dimensão são armazenadas em tabelas separadas de dimensão com uma chave primária. Na tabela de fato é colocada uma chave estrangeira que se associa a uma chave primária correspondente em uma tabela de dimensão, deste modo as dimensões ficam associadas ao fato. Esse modelo de dimensional é conhecido como modelo estrela (*star schema*) [KIM1888].

O esquema estrela apresenta dificuldades quando utiliza muitas dimensões ou quando estas são muito grandes. Na presença de tais características os sistemas não devem forçar a utilização deste esquema [POE1998]. Além disso, este esquema não apresenta uma forma clara de tratar hierarquias implícitas.

O nome "Estrela" está associado à disposição física do modelo, que consiste de uma tabela central, a tabela de fatos, que se relaciona com "n" tabelas de dimensões. A figura 10 apresenta este esquema.

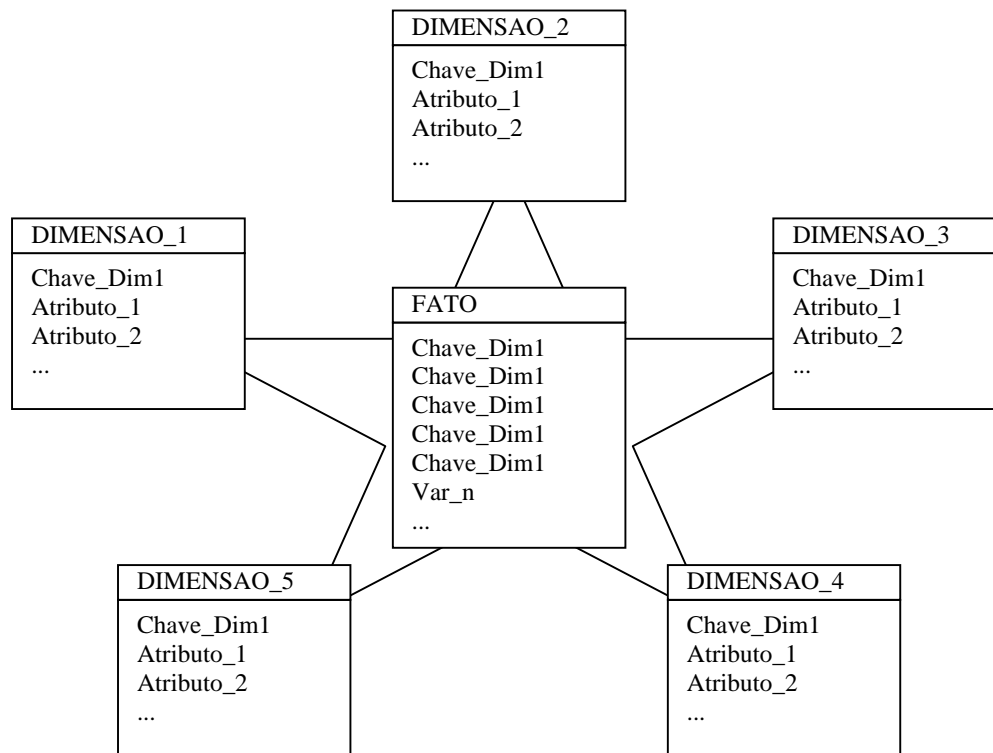


Figura 10: Representação do Esquema Estrela
 Fonte: Adaptado de Kimball, 1998 [KIM1998]

A representação mais simples de um modelo dimensional contém um esquema estrela com uma tabela de fatos relacionada com tabelas de dimensões. Na verdade, um modelo dimensional pode ser representado por uma ou mais tabelas de fatos, relacionadas com tabelas de dimensões. Entretanto, a visão de um esquema por vez torna o modelo mais claro. A seguir serão mostradas as definições e características dos componentes do esquema estrela [RUB1997, MEL1997 e KIM1998].

- Tabela de Fatos

A tabela de fatos representa as informações que serão avaliadas, sendo, normalmente, constituída de valores numéricos que representam os objetos da análise, como por exemplo, total de vendas, total de movimentação e média de reservas canceladas. Algumas vezes é possível encontrar tabelas de fatos sem valores numéricos, nesses casos, normalmente, a tabela de fatos é empregada para mapear

eventos [KIM1998]. A tabela de fatos normalmente é grande, apresentando muitos registros. Esta tabela contém as informações básicas do nível de transação do negócio, de interesse particular a uma aplicação. Uma característica importante da tabela de fatos é a esparsidade, dessa forma, quando não existe valores para um cruzamento de dimensões, não são armazenados zeros. A tabela de fatos armazena as medições numéricas de interesse para o negócio. Os projetistas devem dar preferência aos atributos que representem valores perfeitamente aditivos [KIM1998].

- Classificação dos Atributos Numéricos em uma Tabela de Fatos

Os atributos mais comuns em uma tabela de fatos são valores numéricos. Estes valores podem ser de três tipos:

- a) Valores Aditivos: são valores da tabela de fatos sobre os quais podem ser aplicadas as operações de soma, subtração e média. Os valores, como por exemplo, "total de vendas" e "total de itens vendidos", por *Produto*, por *Região* e por *Loja* representam valores aditivos.
- b) Valores Não Aditivos: são valores da tabela de fatos que não podem ser manipulados livremente, como valores percentuais ou relativos. Para esses tipos de valores, os cálculos devem ser realizados sobre os dados absolutos nos quais se baseiam. Todos os valores que medem um nível de intensidade, são valores estáticos não aditivos. Esses valores são válidos para o momento em que a informação foi obtida, e sua soma através do tempo não tem significado, entretanto, podem ser úteis para futuras manipulações.
- c) Valores Semi Aditivos: são valores que envolvem contagem dupla. Portanto, são restritos a uma dimensão. Quando a análise é efetuada sobre a dimensão aditiva, as operações normais podem ser aplicadas sobre o valor. Segundo Kimball [KIM1998], todas as medições que registram um nível estático, como níveis de estoque e saldos de contas financeiras, e medições de intensidade, como de temperatura ambiente, são informações não aditivas ao longo do tempo. Entretanto, podem ser agregadas, de forma útil, ao longo do

tempo através do cálculo da média do número de períodos de tempo. Portanto, podem ser considerados valores semi-aditivos.

- Tabela de Dimensão

A tabela de dimensão armazena as informações necessárias para análises ao longo de dimensões, sendo normalmente, menor que a tabela de fato. Esta tabela apresenta chave simples e seus campos, normalmente descritivos, são empregados como fonte das restrições e linhas de cabeçalhos para relatórios.

A qualidade do banco de dados é proporcional a dos atributos de dimensões, portanto deve ser dedicado tempo e atenção a sua descrição, ao seu preenchimento e a sua garantia de qualidade [KIM1998].

- Hierarquia de Dimensões

As hierarquias são a base para a agregação de dados e para a navegação entre os diferentes níveis de detalhe em uma estrutura multidimensional [THO1997] [MEY1998]. Segundo Thomsen [THO1997], "uma hierarquia é um atributo de uma dimensão". As hierarquias descrevem a estrutura organizacional e lógica dos relacionamentos entre os dados [MEY1998], por exemplo: *categoria / tipo / produto*. Muitas dimensões apresentam uma estrutura hierárquica ou multinível. Como exemplo, pode-se citar a dimensão Tempo, que pode ser definida por *ano / mês / dia*.

Na segunda fase da modelagem é construído o modelo físico, onde são incluídas características físicas e chaves. Tendo definido os atributos que figurarão nas tabelas de fatos e nas dimensões, resta ainda decidir sobre importantes aspectos de organização dos dados e performance, representados inicialmente pela definição da granularidade, do particionamento, e da definição de agregados. A seguir serão explicadas as definições de granularidade, particionamento e agregados.

- Granularidade

Para Inmon, a granularidade é a principal questão de projeto, pois afeta profundamente o volume dados no Data Warehouse e o tipo de consulta a ser oferecida. O nível de detalhe de uma consulta vai determinar o balanceamento do nível de granularidade contido nas unidades de dados no DW, e conseqüentemente o volume de dados. Quanto mais detalhe, mais baixo o nível de granularidade. Quanto menos detalhe, mais alto o nível de granularidade

A maioria das organizações encontra grandes dificuldades no balanceamento da granularidade, sendo que a melhor solução consiste em alguma estrutura de vários níveis de granularidade. O balanceamento no gerenciamento da questão da granularidade deve ser feito cuidadosamente no início do projeto e construção do Data Warehouse [INM1997].

Segundo Kimball, na maioria das vezes é preciso expressar os dados no menor nível de granularidade, para que as consultas possam aprofundar-se com precisão no Data Warehouse. A integração de duas fontes normalmente significa expressá-las no menor nível de granularidade comum a ambas as fontes. Por exemplo uma fonte baseada em meses e outra em semanas, a alternativa mais eficiente é integrar as fontes no nível diário [KIM1998].

Singh, afirma que sob muitos aspectos utilizar o maior nível de detalhamento representa segurança, mesmo com a disponibilidade de outros níveis sumarizados.

Quanto maior for o nível de sumarização, maior a eficiência e a velocidade de obtenção dos dados, ou seja, os analistas de SAD manipulam mais os dados sumarizados [SIN2001].

A maioria das organizações precisa dispor de dois níveis de granularidade no Data Warehouse: o nível baixo de granularidade em função da segurança e da precisão obtida, como também o nível mais alto de granularidade em função da velocidade e eficiência na obtenção dos dados [INM1997, KIM1998 e SIN2001].

- Particionamento

Para Inmon, depois da granularidade outra questão importante do projeto dos dados é o particionamento de dados. O particionamento de dados é a separação dos dados de detalhe corrente em unidades físicas separadas que podem ser tratadas de forma independente, por serem menores permitem maior flexibilidade no gerenciamento dos dados [INM1997].

Segundo Singh, o particionamento pode ser feito no nível do SGBD e/ou no nível da aplicação. No particionamento no nível SGBD, devido ao conhecimento das partições, o gerenciamento e a infra-estrutura ficam a cargo do SGBD. No nível de aplicação o particionamento é feito pelo código da aplicação controlado pelo programador, o que possibilita maior flexibilidade de gerenciamento de dados [SIN2001].

Inmon, chama o nível do SGBD de nível de sistema. Para ele esse nível pressupõe uma definição de dados única, o que dificulta para o Data Warehouse que trabalha com horizontes temporais de até 10 anos, a definição dos dados, normalmente, é alterada com o passar do tempo. No nível de aplicação, o SGBD e o sistema operacional (SPT/OLTP) não tomam conhecimento do particionamento, sendo possível existir uma definição de dados diferente para cada ano [INM1997].

- Agregados

Segundo Kimball [KIM1998], “Agregado é um registro de tabela de fatos que representa o resumo dos registros de nível básico da tabela de fatos”. Em outras palavras, sumários dos dados contidos no modelo original, organizados de forma a atender consultas rotineiras de forma mais ágil. Constitui uma das formas mais eficientes de melhorar a performance das consultas sobre o Data Warehouse [NEI1997].

Embora a definição de agregados ocorra mais freqüentemente durante a utilização por parte dos usuários, é válido projetá-los já na fase de definição do

modelo de dados, para orientar dimensionamento de *hardware* e ajustes no próprio modelo, aproveitando também que os requisitos do usuário acabaram de ser definidos.

Deverão ser considerados candidatos os agregados que reduzirem consideravelmente o volume em relação ao modelo original, normalmente em torno da razão de 10 ou mais [DON1997].

Existem duas abordagens para armazenar agregados: definindo novas tabelas de fatos ou definindo campos nível. A utilização da primeira abordagem torna mais simples a manutenção, a carga e a utilização dos dados [KIM1998].

2.4.3 Preparação e Desenvolvimento de Ferramentas de Extração

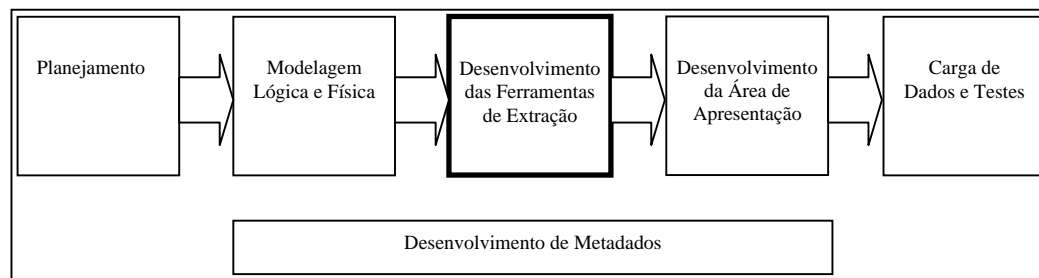


Figura 11: Desenvolvimento das Ferramentas de Extração

Extração é o processo de coleta dos dados existentes nos sistemas operacionais e sistemas externos para incorporação no Data Warehouse. Os dados são coletados, tratados e transformados no formato do modelo de dados do Data Warehouse e finalmente incorporados na base de dados.

A definição e implementação da extração são de suma importância em um projeto de Data Warehouse, consumindo muito tempo para serem realizadas, tendo em vista a complexidade. A figura 11 localiza a fase de desenvolvimento das ferramentas de extração dentro das fases típicas do desenvolvimento do Data Warehouse e a figura 12 representa o processo de extração de dados dos sistemas operacionais.

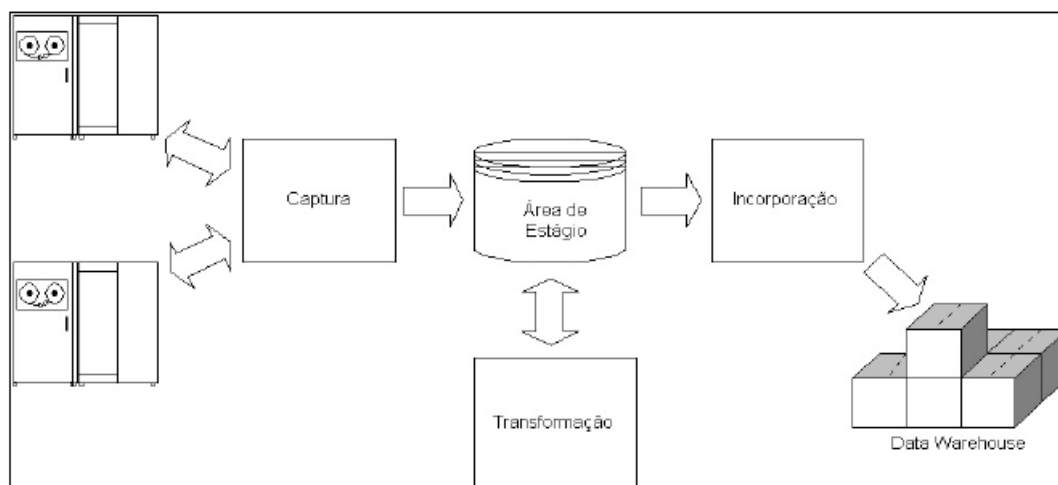


Figura 12: Processo de Extração
 Fonte: Adaptado de CHUCK, 1998.

a. Coleta

É muito comum que os dados nos sistemas operacionais não sejam integrados. Também freqüente é o fato de que os mesmos dados sejam representados de maneira diferente entre os diversos sistemas operacionais. Segundo Inmon [INM1997], as principais diferenças encontradas na representação são:

- nomenclatura diferente para identificação do mesmo dado. Exemplo: filiação e nome do pai;
- mesma nomenclatura, mas definição diferente. Exemplo: no sistema A o sexo é definido como um caracter contendo “M” ou “F”. Já no sistema B, o sexo é definido pelo número “0” ou “1”;
- unidades de medida diferentes. Exemplo: no sistema A, a dimensão de um objeto é medida em centímetros, já no sistema B a dimensão é medida em polegadas.

Tendo os sistemas definições diferentes, é preciso que ocorra uma conversão dos dados de acordo com as definições do modelo de dados do Data Warehouse.

Mas além da representação dos dados de forma diferente, há ainda a preocupação com a plataforma em que os sistemas operacionais são mantidos. Serão

necessários procedimentos especiais para conversão dos dados mantidos nestes para a plataforma em que o Data Warehouse será mantido.

O ciclo de dados num Data Warehouse define o período de atualização dos mesmos. O ciclo varia de acordo com a natureza dos dados, podendo ser diário, mensal, trimestral ou anual.

Quanto aos tipos de carga dos dados Inmon [INM1997] cita três tipos, sendo:

- carregamento de dados históricos;
- carregamento de dados de valor corrente no ambiente operacional;
- carregamento de alterações do Data Warehouse. a partir de atualizações que tenham ocorrido nos sistemas operacionais.

Quanto às técnicas Inmon [INM1997] cita cinco tipos sendo:

- pesquisa de dados por datas;
- pesquisa em um arquivo de informações sobre alterações desde a última carga;
- pesquisa nos arquivos de *log* (históricos);
- modificação no código dos sistemas operacionais de forma que estes, ao registrarem alterações, atualizem de forma automática o Data Warehouse;
- comparação entre arquivos de imagem, sendo comparado serialmente o arquivo de imagem retirado após a última extração com o arquivo de imagem tirado no momento da nova extração.

b. Transformação

No processo de transformação, os dados coletados são tratados em uma área intermediária conhecida como área de estágio. Em geral, é necessário construir ou adquirir uma ferramenta personalizada para realizar a transformação (bem como o processo de coleta e incorporação) que, além de ser capaz de lidar com a heterogeneidade dos bancos de dados e dos próprios dados, tenha as seguintes funcionalidades [BER1997]:

- remoção de dados indesejáveis;
- conversão para nomes e definições comuns dos dados;
- cálculo de dados personalizados e derivados;
- estabelecimentos de valores *default* para dados esquecidos;
- documentação da origem das mudanças de definições de dados no metadados;
- geração de chaves.

c. Incorporação

O processo de incorporação consiste da carga no Data Warehouse dos dados mantidos na área de estágio, criada no processo de transformação. Existem três técnicas básicas de incorporação [CHU1998]:

- alteração: nesta técnica os dados existentes nas tabelas do Data Warehouse são alterados pelos gerados pelo processo de transformação. Se as tabelas existentes não existem, serão criadas;
- inserção: nesta técnica os dados existentes na área de estágio serão inseridos às tabelas existentes no Data Warehouse sem sobrepor os existentes;
- inserção construtiva: nesta técnica os dados existentes na área de estágio serão inseridos nas tabelas do Data Warehouse e serão atualizados valores calculados que representam estados.

2.4.4 Desenvolvimento de Ferramentas de Aplicações Cliente

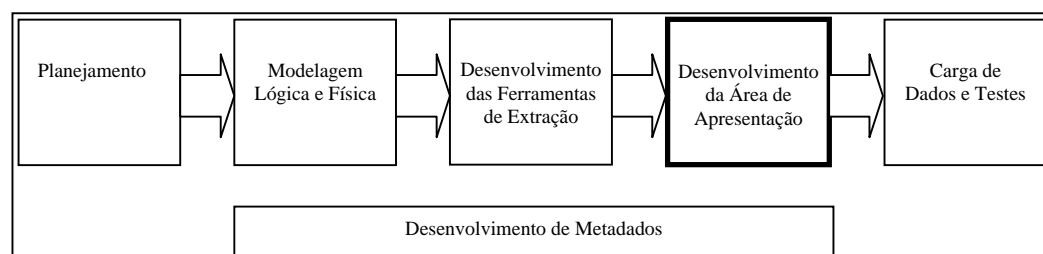


Figura 13: Desenvolvimento da área de apresentação

O valor do Data Warehouse pode ser medido sob o ponto de vista das informações que possam ser extraídas, visto que toda a arquitetura até aqui apresentada tem por fim a disponibilização de informações estratégicas no apoio a decisões. Dentre as formas que podem ser utilizadas para interagir com o Data Warehouse vale citar: OLAP, análises estatísticas e *data mining*. A figura 13 localiza a fase de desenvolvimento da área de apresentação dentro das fases típicas do desenvolvimento do Data Warehouse.

Segundo Harrison [HAR1997], o rótulo OLAP (On-Line Analytical Processing ou Processamento Analítico On-Line) foi reservado para análises multidimensionais. As consultas simples e relatórios representam as facilidades mais simples proporcionadas pelo OLAP. Tipos mais sofisticados de consultas permitem que o usuário “navegue” por níveis diferentes de resumo dos dados, até chegar ao nível de detalhe encontrado nos sistemas operacionais ou ainda combinar dados na mesma consulta para a detecção de relações entre dados, de forma multidimensional.

Uma operação muito útil realizada pelos OLAPs é o chamado *drill-down*. Ao ser realizado um *drill-down* aumenta-se o nível de detalhe da informação ao longo de uma dimensão. Supondo uma consulta que resultou num relatório contendo o total de mortos por região, pode ser evidenciado um número elevado de mortes em uma região específica. Ao ser realizado um *drill-down* sobre esta região, o total de mortos é visualizado para cada município que a compõe. A operação de *drill-up* percorre o caminho inverso, ou seja, do dado específico para o geral.

Quando o rótulo OLAP foi introduzido, os dados eram extraídos dos bancos de dados relacionais e carregados em bancos multidimensionais. O dado, então, era analisado através das funcionalidades citadas com extrema rapidez. Este método foi denominado MOLAP ou OLAP Multidimensional. Em seguida outro método denominado ROLAP ou OLAP Relacional foi introduzido com muito sucesso, devido à facilidade de mapeamento por utilizar o modelo relacional como base para as análises, e dispensar a utilização de um banco de dados multidimensional proprietário [DON1997].

As análises estatísticas empregadas vão além da simples redução de dados a serem apresentados, como cálculos de média. HAR1997 citou que análises estatísticas mais sofisticadas incluem emprego de métodos de regressão para estabelecimento da relação entre variáveis dependentes (venda de produtos) com variáveis independentes (como preço e clima), e outros métodos.

Data mining ou mineração de dados pode ser conceituada como um processo de reorganização e análise dos dados para a descoberta de informações ocultas. O *data mining* utiliza técnicas de análises estatísticas bem como técnicas de inteligência artificial, como redes neurais, algoritmos genéticos, reconhecimento de padrões, conjuntos difusos, etc [BER1997].

2.4.5 Metadados

Os metadados são dados sobre os dados do Data Warehouse, constituindo assim o componente mais importante de sua estrutura. Os metadados não contêm dados retirados diretamente do ambiente operacional, portanto se situam em uma dimensão diferente dos outros dados do DW. A figura 14 localiza a fase de desenvolvimento de metadados dentro das fases típicas do desenvolvimento do Data Warehouse.

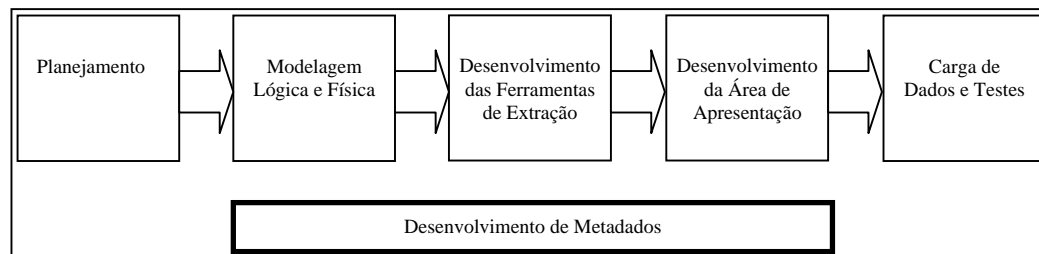


Figura 14: Desenvolvimento de Metadados

Os metadados contêm, pelo menos, as seguintes informações [INM1997]:

- A estrutura dos dados;
- Os Algoritmos usados para a sintetização;
- A fonte de dados que alimenta o DW;
- O mapeamento desde o ambiente operacional até o DW;

- O modelo de dados;
- O relacionamento entre o modelo de dados e o DW;
- O histórico de extrações.

É necessário entender quais os dados que estão disponíveis no Data Warehouse e onde estão localizados, para serem acessados com eficiência. Os metadados constituem um catálogo dos dados do DW e os indicadores/ponteiros para esses dados. Os metadados também podem conter as seguintes informações [SIN2001]:

- Estatística de uso de dados;
- Medições de desempenho;
- Diferentes elementos por atributo.

As informações contidas nos metadados têm um papel muito importante no Data Warehouse e são utilizadas de diversas maneiras [INM1997]:

- No auxílio dos analistas de SAD na localização dos dados do DW;
- Como guia para o mapeamento dos dados em sua transformação desde o ambiente operacional até o DW;
- Como guia para os algoritmos utilizados para sintetização entre os dados detalhados atuais e os dados altamente resumidos; etc.

2.5 METODOLOGIAS DE DESENVOLVIMENTO

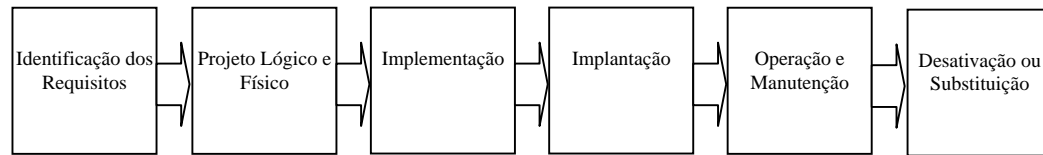
Várias são as metodologias utilizadas para a implementação de sistemas de informações e variadas são as formas de como as mesmas são utilizadas. Apesar de muitos anos de desenvolvimento destes sistemas, não existe um consenso entre os desenvolvedores.

As variáveis afetadas pela escolha da metodologia são entre outras, o prazo, o retorno do investimento, satisfação do usuário, quantidade de implementação a ser refeita, requisitos de recursos e flexibilidade para adequações [CHU1998].

Existem correntes de desenvolvedores que defendem o desenvolvimento dos sistemas de informações utilizando a abordagem tradicional de desenvolvimento de sistemas operacionais. Nesta metodologia existem fases bem definidas e cada qual

termina após um determinado número de iterações. A figura 15 apresenta o ciclo de vida do desenvolvimento tradicional.

Figura 15: Ciclo de Vida Tradicional de Desenvolvimento



Fonte: Adaptado de LAUDON, 1999 [LAU1999]

O ciclo de vida tradicional de concepção de sistemas é baseado na filosofia de “projeto”, onde existe um começo, meio e fim, com pouca ênfase sobre continuidade e refinamento. Nesta abordagem, normalmente no projeto lógico da organização de dados é adotado o modelo de Entidade-Relacionamento altamente normalizado. Esta abordagem vem sendo utilizada há muitos anos e apresenta alguns problemas devido à sua estrutura rígida. O quadro 2 apresenta a relação das principais vantagens e desvantagens da metodologia.

Quadro 2: Ciclo de desenvolvimento orientado ao projeto

Vantagens	Desvantagens
Técnicas de Modelagem dominadas pelo corpo técnico	Dificuldade na realização de ajustes para adequação no modelo de dados, devido à forte dependência da implementação sobre o modelo.
	Fragilidade quanto a mudanças de cenários e necessidades dos usuários, devido a pouca previsão de novos ciclos de análise e implementação.
	A utilização do modelo entidade-relacionamento normalmente não é adequada a um sistema de informações devido à falta de clareza, performance inferior das consultas por conta da necessidade de efetuar-se muitas junções entre tabelas e ainda dificuldade em realizar-se mudanças na estrutura.

A metodologia baseada no Data Warehouse global, que foi utilizada neste capítulo para apresentar as fases existentes no Data Warehousing, é orientada

especificamente para sistemas de informações, devido à utilização da técnica de Data Warehouse, bem como as metodologias que serão apresentadas em seguida.

Esta abordagem parte da idéia de desenvolvimento *Top-down* (de cima para baixo). Inicialmente serão levantados todos os requisitos, fontes de dados, padrões, e outros, dos departamentos da empresa participantes da iniciativa, para então dar seqüência na implementação de toda a estrutura.

Apesar de ser orientado para sistemas de informações, esta abordagem apresenta algumas sérias desvantagens, mas também apresenta vantagens, como descritas no quadro 3.

Quadro 3: Metodologia *Top-down*

Vantagens	Desvantagens
Como as fontes de dados são acessadas uma única vez e armazenadas em um único repositório, existe uma grande probabilidade de que os dados mantidos no <i>Data Warehouse</i> sejam dados confiáveis e coerentes, com representações unívocas.	O processo de análise e triagem de todos os dados existentes, análise de regras de negócio e todos os relacionamentos possíveis é demasiadamente custoso e demorado. Assim, o projeto só retornará algum resultado após muitos meses, tornando-se politicamente difícil de gerenciar.
Devido ao desenvolvimento da extração de dados ser feito uma única vez, evita-se replicação de esforços.	Os gastos iniciais com aquisição do <i>hardware</i> e infra-estrutura e formação da numerosa equipe técnica são demasiadamente elevados.
	O projeto e a implementação do processo de extração são complexos devido à enorme diversidade inicial de dados.

Devido ao fracasso de inúmeras implementações de Data Warehouse globais, empresas de consultoria apresentaram uma abordagem diferenciada para o desenvolvimento de Data Warehouse. Esta metodologia defende a construção de pequenos conjuntos do Data Warehouse baseados em assuntos específicos (*data marts*), para a apresentação de resultados em tempo bem mais reduzido [VAS1999].

Ao contrário da abordagem anterior, esta parte do desenvolvimento *Bottom-up* (de baixo para cima), baseada em um processo contínuo de desenvolvimento. Inicialmente, é realizada uma análise dos requisitos de forma global, de onde surge

uma lista de data marts a serem implementados e como serão integrados. A partir daí, serão levantados os requisitos de um dos departamentos integrantes da iniciativa e implementado o data mart correspondente. O ciclo repete-se, incrementalmente, até que todos os data marts tenham sido implementados. O conjunto de todos os data marts desenvolvidos constitui o Data Warehouse da empresa.

A figura 16 ilustra o processo de desenvolvimento desta abordagem [KIM1998] e no quadro 4 são retratadas suas vantagens e desvantagens.

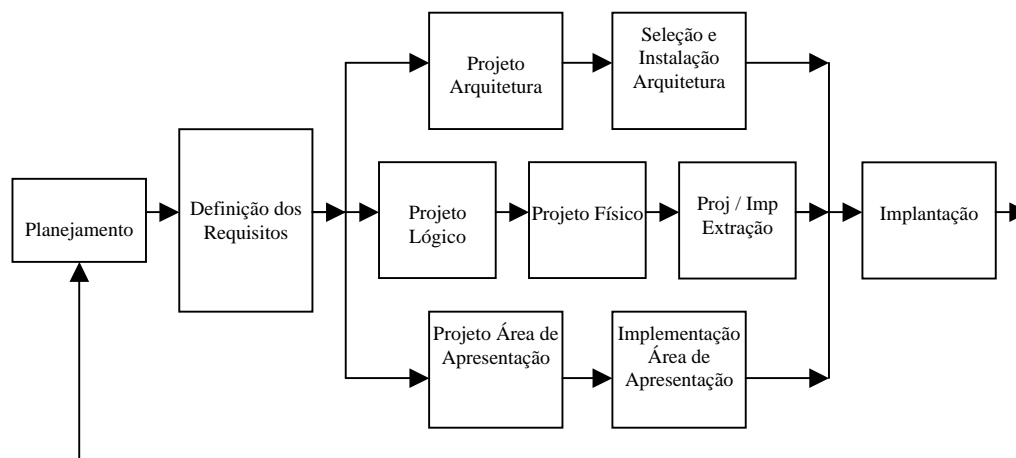


Figura 16: Etapas do Desenvolvimento com Data Marts Incrementais

Quadro 4: Desenvolvimento com Data Marts Incrementais

Vantagens	Desvantagens
A apresentação dos primeiros resultados é feita de modo mais rápido e barato do que a abordagem global	Complicações políticas por conta da determinação da seqüência de implementação dos data marts e das prioridades de manutenção
A integração entre os data marts possibilita a unicidade de representação dos dados e informações mais confiáveis por não existirem redundâncias	Metadado mais complexo para gerenciar a distribuição e integração dos dados
Os mecanismos de extração são projetados uma única vez	Maior controle no nível de granularidade, padronização e nas manutenções das tabelas compartilhadas

Dentre as metodologias apresentadas, considera-se mais adequada a baseada em data marts incrementais, devido principalmente ao retorno rápido e à maior perspectiva de integração dos dados. Em seguida será apresentada a arquitetura proposta para a implantação em conjunto com a metodologia baseada em data marts incrementais.

CAPÍTULO 3

ABORDAGEM PROPOSTA PARA O DESENVOLVIMENTO INCREMENTAL DE UM DATA WAREHOUSE

A escolha da arquitetura e metodologia de desenvolvimento deve ser a primeira decisão a ser tomada antes de se iniciar o desenvolvimento do sistema de informação. A escolha pode ser modificada após o início da implementação, mas normalmente esta decisão acarretará em grande volume de trabalho a ser feito.

Neste capítulo serão abordadas as principais características das arquiteturas empregadas em sistemas de informações baseados em Data Warehouse e a arquitetura proposta. Em seguida será indicado um conjunto de técnicas para se estabelecer uma metodologia de desenvolvimento que permitam uma modelagem incremental do ambiente de Data Warehousing.

O conjunto de técnicas proposto na metodologia de desenvolvimento utiliza, em um regime de colaboração, duas abordagens de modelagem existentes na literatura: o emprego de técnicas de modelagem dimensional, com base na especificação do usuário final, para o desenvolvimento de Data Warehouse e data marts [KIM1998]; e o desenvolvimento de Data Warehouse a partir do modelo corporativo da empresa [SIL1997].

3.1 ARQUITETURAS

Conforme se apresentou no capítulo anterior, os componentes comuns existentes em um sistema de informação baseado em Data Warehouse são distribuídos em três áreas, sendo estas a área de extração, área de armazenamento e área de apresentação. As diferenças existentes entre as arquiteturas são resumidas à forma de implementação das áreas, principalmente da área de apresentação e de armazenamento.

Na área de extração, a maioria das arquiteturas possui um único componente de extração que é responsável pela coleta dos dados operacionais para um única área de estágio, além de efetuar a limpeza, transformação e inserção no Data Warehouse valendo-se do metadados [KIM1998, INM1997, TAN1998 e DEB1999].

Na área de armazenamento, algumas arquiteturas divergem quanto ao escopo e quanto à forma de organização dos dados. Dentre as formas mais comuns de organização de dados, podem-se destacar:

- Data Warehouse global: a área de armazenamento é composta por um único repositório (particionado ou não) que reúne os dados necessários a suportar toda ou grande parte da corporação. Deste repositório podem ser derivados data marts [INM1997 e CHU1998];
- Data marts isolados: a área de armazenamento é composta por vários *data marts* projetados de forma isolada, sem interligação [VAS1999 e CHU1998], denominados *Legamarts* ou data marts legados;
- Data marts integrados: a área de armazenamento é composta por vários data marts projetados de forma a manter a integração dos dados. A interligação dos dados faz-se através do metadados e pelo software gerenciador de banco de dados [KIM1998 e CHU1998].

Na área de apresentação, a principal diferença encontrada na literatura se dá quanto à forma e ao local onde as requisições das aplicações clientes são processadas. As formas mais comuns são:

- conexão direta: as aplicações clientes conectam-se diretamente ao Data Warehouse para realizar as consultas e a lógica encontra-se nas próprias aplicações clientes apoiadas ou não por um metadados mantidos externamente [TAN1998 e KIM1998];
- conexão por servidor: as aplicações clientes submetem requisições de dados ao servidor (OLAP, WEB, servidor de aplicações, entre outros), no qual reside a lógica do processamento. O servidor então se vale do metadados para processar a consulta para em seguida enviar o resultado à aplicação cliente requisitante. No caso do servidor OLAP, as requisições de dados são processadas junto ao cubo do servidor ou

junto ao Data Warehouse em caso de necessidade [TAN1998] [KIM1998].

3.1.1 Tipos de Arquiteturas

Várias são as arquiteturas descritas na literatura e propostas por empresas de consultoria especializada. As principais diferenças entre as arquiteturas baseiam-se na forma de organização da área de armazenamento e da área de apresentação, conforme apresentado anteriormente. Entre as principais arquiteturas figuram a *Top-down*, a *Bottom-up* e a Intermediária.

- Arquitetura Top-Down

Nessa arquitetura o processo se inicia com a criação de um *Data Warehouse* e depois se parte para a segmentação, ou seja, divide-se o *DW* em áreas menores gerando assim pequenos bancos orientados por assunto departamentalizados, ou seja, os data marts.

O *Data Warehouse* integra todos os dados num formato comum e num ambiente comum de software. Todos os recursos de uma organização e todos os dados necessários para o suporte à decisão são consolidados na construção do *DW*. O quadro 5 aponta as principais vantagens e desvantagens da arquitetura *Top-down*.

Este tipo de arquitetura tipicamente utiliza uma extensão do Data Warehouse no ambiente dos sistemas operacionais, chamadado ODS (*Operational Data Store*). O ODS possui dados mais recentes que o Data Warehouse, tem menor volume e pode ser atualizado (volátil). Geralmente o ODS é utilizado para o refinamento do modelo de dados e pode servir de carga para o DW. A arquitetura *Top-down* utilizando o ODS pode ser representada pela figura 17.

Quadro 5: Arquitetura Top-Down

Vantagens	Desvantagens
Herança de arquitetura - Todos os data marts originados a partir de um DW, utilizam a arquitetura e os dados deste DW, permitindo uma fácil manutenção	Implementação muito longa - Os DW são, normalmente, desenvolvidos de modo iterativo, por áreas de assuntos, como por exemplo, vendas, finanças e recursos humanos. Mesmo assim, é necessário muito tempo para que a primeira área de assunto entre em produção, dificultando a garantia de apoio político e orçamentário
Visão de empreendimento - O DW concentra todos os negócios da empresa, sendo possível a partir dele extrair níveis menores de informações	Alta taxa de risco - Não existem garantias para o investimento neste tipo de ambiente
Repositório de metadados centralizado e simples - O DW provê um repositório de metadados central para o sistema. Esta centralização permite manutenções mais simples do que aquelas realizadas em múltiplos repositórios	Heranças de cruzamentos funcionais. É necessário uma equipe de desenvolvedores e usuários finais altamente capacitados para avaliar as informações e consultas que garantam à empresa habilidade para sobreviver e prosperar na arena de mudanças de competições políticas, geográficas e organizacionais
Controle e centralização de regras - A arquitetura top-down garante a existência de um único conjunto de aplicações para extração, limpeza e integração dos dados, além de processos centralizados de manutenção e monitoração.	Expectativas Relacionadas ao Ambiente - A demora do projeto e a falta de retorno pode induzir expectativas nos usuários.

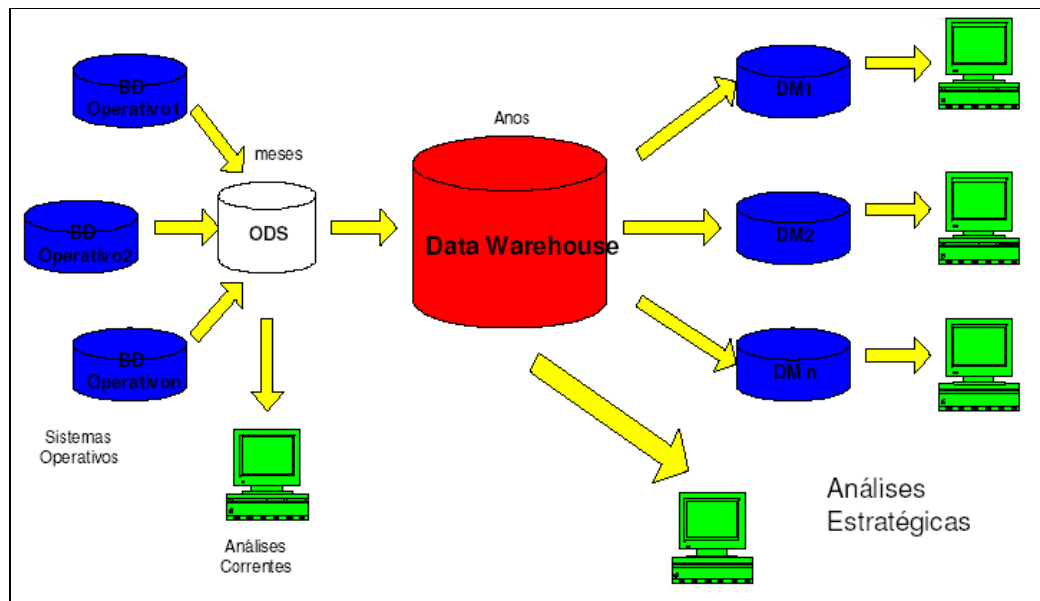


Figura 17: Arquitetura Top-Down
 Fonte: SANTOS, 2005 [SAN2005]

- Arquitetura Bottom-Up

Nesta arquitetura são desenvolvidos data marts independentes que podem se transformar no Data Warehouse ou não. A situação é inversa à estratégia *Top-Down*.

Muitas organizações preferem primeiro criar um banco de dados para somente uma área ou departamento. Com isso os custos são bem inferiores de um projeto de Data Warehouse completo. A partir da visualização dos primeiros resultados partem para outra área assim sucessivamente até resultar num DW. A arquitetura proposta por Kimball [KIM1998] de data marts dimensionais que são interligados através do Data Warehouse, onde estão as tabelas de dimensões e fatos conformados, permite a conexão dos data marts formando um DW corporativo. A figura 18 apresenta a arquitetura *Bottom-up*, em seguida o quadro 6 indica as principais vantagens e desvantagens desta arquitetura.

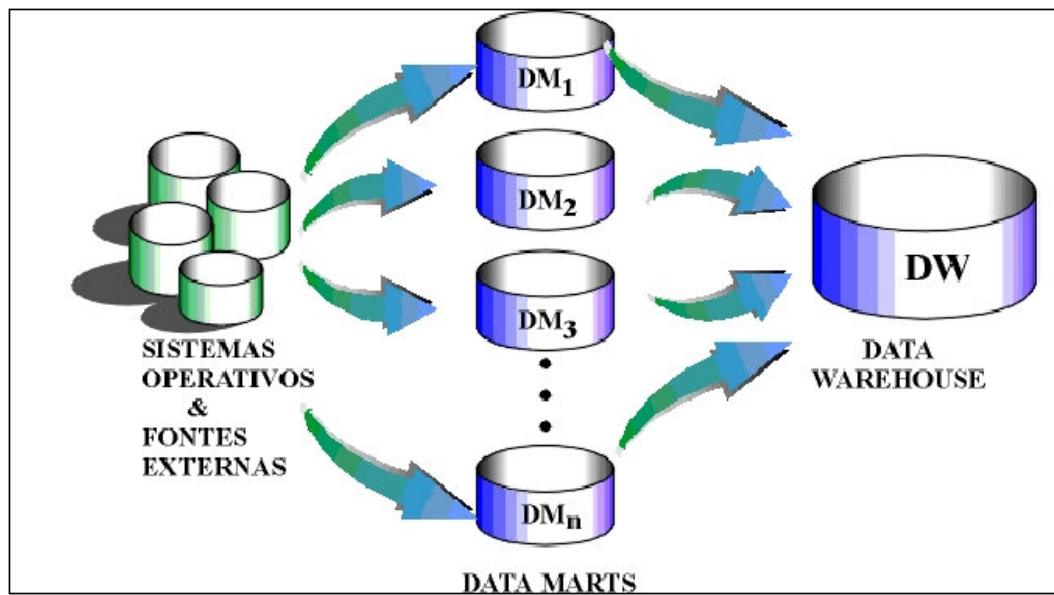


Figura 18: Arquitetura *Bottom-Up*
Fonte: SANTOS, 2005 [SAN2005]

Quadro 6: Arquitetura *Bottom-Up*

Vantagens	Desvantagens
Implementação rápida - A construção dos data marts é altamente direcionada, permitindo um rápido desenvolvimento.	Perigo de LegaMarts - Um dos maiores perigos no ADW é a criação de data marts independentes. O advento de ferramentas de "drag-and-drop" facilitou o desenvolvimento de soluções individuais, de acordo com as necessidades da empresa. Estas soluções podem não considerar a arquitetura de forma global. Desta forma, os data marts independentes transformam-se em data marts legados, ou LegaMarts. Os LegaMarts dificultam, quando não inviabilizam futuras integrações. Eles são parte do problema e não da solução
Retorno Rápido - A arquitetura baseada em dat marts com incremento demonstra rapidamente seu valor, permitindo uma base para investimentos adicionais, com um nível mais elevado de confiança	Desafio de Possuir a Visão de Empreendimento - Durante a construção dos data marts incrementais é necessário que se mantenha um rígido controle do negócio como um todo. Este controle requer um maior trabalho ao extrair e combinar as fontes individuais do que utilizar um Data Warehouse
Manutenção do Enfoque da Equipe - Um dos maiores desafios do desenvolvimento de um ambiente de Data Warehouse é a manutenção do mesmo enfoque por toda a equipe. A elaboração de data marts incrementais, permite que os principais negócios sejam enfocados inicialmente, sem que haja gastos no desenvolvimento de áreas que não são essenciais ao problema	Administrar e Coordenar Múltiplas Equipes e Iniciativas - Normalmente, esse tipo de arquitetura emprega o desenvolvimento de data marts em paralelo. Isto pode conduzir a uma rígida administração tentando coordenar os esforços e recursos das múltiplas equipes, especialmente nas áreas de regras e semântica empresariais
Herança Incremental - A estratégia de data marts incrementais obriga a entrega de recursos de informação, passo a passo. Isto permite a equipe crescer e aprender, reduzindo os riscos. A avaliação de ferramentas, tecnologias, consultores e vendedores só deve ser realizado uma vez, a não ser que existam restrições que impeçam o reaproveitamento	

- Arquitetura Intermediária

Segundo Hackney [HAC1998], esta arquitetura tem a finalidade de integrar a estratégia *Top-down* com a *Bottom-up*. Nesta abordagem efetua-se a modelagem de dados do Data Warehouse, sendo o passo seguinte a implementação de partes desse modelo. Essas partes são escolhidas por área de interesse e constituem os data marts. Cada data mart gerado a partir do modelo de dados do Data Warehouse é integrado no modelo físico do DW. A principal vantagem desta estratégia é a garantia de consistência dos dados. Essa garantia é obtida em virtude do modelo de dados para o DW ser único possibilitando realizar o mapeamento e o controle dos dados.

3.2 A ARQUITETURA PROPOSTA

Um dos principais fatores para o fracasso de muitos sistemas é o enfoque orientado a “projeto”. Em um ciclo de vida de software orientado a “projeto”, encara-se o desenvolvimento como uma seqüência rígida de tarefas, com pouco enfoque na criação de uma arquitetura flexível a mudanças. [SEL2001]

Devido à flexibilidade e ao foco num processo evolutivo contínuo, a utilização de uma metodologia de desenvolvimento baseada em Data Warehousing orientada a “processo”, viabiliza o desenvolvimento de sistemas de informações permitindo o acompanhamento das mudanças de necessidades dos tomadores de decisões.

Com a utilização de uma metodologia de desenvolvimento baseada em Data Warehouse orientado a “processo”, torna-se possível o uso de uma metodologia de Data Warehouse baseado em um processo contínuo de desenvolvimento e adequação às necessidades do usuário. Para a implementação de um Data Warehouse na forma de processo, verifica-se que o modelo de data mart incremental definido pela arquitetura *Bottom-up* é o mais indicado, devido ao seu enfoque incremental e integrado. De acordo com o surgimento de novas necessidades, novos data marts podem ser implementados ou ajustes podem ser realizados nos existentes. A arquitetura distribui os componentes físicos do Data Warehouse em diversos

módulos, o que permite grande flexibilidade e rapidez no desenvolvimento, possibilitando um fácil acompanhamento das necessidades dos usuários, além de outros benefícios.

3.3 METODOLOGIA DE DESENVOLVIMENTO ADOTADA

As metodologias apresentadas na literatura dão grande importância à modelagem do ambiente de Data Warehouse. Porém, grande parte destas metodologias apresenta o desenvolvimento deste ambiente como uma arquitetura *Top-down* (MEY1998 e INM1998).

Portanto, tendo em vista o acompanhamento constante do usuário, além da rapidez e flexibilidade no desenvolvimento citadas anteriormente, este trabalho busca dar um tratamento metódico a etapa de modelagem, onde cada data mart é integrado e tratado, gerando um pré-modelo que representa o modelo base para a derivação do modelo dimensional. Este esboço é refinado através de técnicas dimensionais, gerando o modelo dimensional final.

Durante o desenvolvimento do caso de uso, este trabalho procura demonstrar as etapas da metodologia de desenvolvimento incremental, baseando-se na tese de mestrado de Vânia Soares [VANIA 1998] e modificando, quando necessário, aquelas que não se adaptarem ao sistema proposto.

A metodologia definida por Soares está representada pela figura 19 e, resumidamente, pode ser definida pelo estabelecimento das seguintes fases:

- **FASE A - ESTUDAR OS MODELOS EXISTENTES:** Nesta fase procura-se definir o escopo dos modelos de dados a serem analisados.

- **FASE B – ELABORAR O PRÉ-MODELO:** Esta fase recebe o modelo definido na fase A e gera o pré-modelo pronto para ser derivado no modelo dimensional. Ela é subdividida em 3 fases:

- **FASE B.I – LIMPEZA E TRANSFORMAÇÃO DO DIAGRAMA ENTIDADE-RELACIONAMENTO (DER) :** Nesta fase é feita a exclusão das informações sem importância para a análise, desnormalizando as relações necessárias, definindo categorias e artefatos, deixando em cada DER apenas as informações de interesse para a solução do problema.
- **FASE B.II – INTEGRAR DER:** Nesta fase é realizada a integração dos DER de cada sistema operativo escolhido, limpos e transformados pela fase anterior. No caso de uso deste estudo, como existirá somente um DER de origem em cada data mart, esta fase não será apresentada.
- **FASE B.III – REFINAR O DER:** Nesta fase ocorre o tratamento do DER da fase anterior transformando-o em pré-modelo. Enquanto a fase anterior preocupa-se com o tratamento conceitual do ER, nesta nova fase é feito o tratamento lógico dos dados. São parte desta fase:
 - Criação de novas chaves, independentes do modelo operativo;
 - Tratamento da periodicidade, fazendo com que o Pré Modelo contemple algumas informações ao longo do tempo não tratadas nos ambientes operativos;
 - Inserção da Chave Tempo para o rastreamento das alterações feitas na base operativa;
 - Estabelecimento de Padrões e Valores Default para os atributos que são derivados de mais de um sistema, podendo estar em formatos diferentes ou estarem ou não preenchidos em certas entidades;
 - Definição de regras para conversão de códigos e abreviações do sistema operativo;
 - Criação de atributos derivados para evitar o cálculo de maneiras diferentes do mesmo valor nos diferentes sistemas.

- **FASE C – LEVANTAMENTO DE FATOS :** A partir do pré-modelo gerado são derivados os modelos dimensionais de cada data mart criado. Esta fase está dividida em duas.

- **FASE C.I – LEVANTAMENTO DE FATOS :** A partir do pré-modelo gerado precisam ser identificados os fatos que farão parte de cada data mart criado.
- **FASE C.II – DERIVAR OS MODELOS DIMENSIONAIS :** Nesta fase é realizada a derivação de cada fato básico estabelecido na fase anterior. Para cada fato escolhido deve-se selecionar a entidade *Fato*, estabelecer o esboço do modelo dimensional através da construção de uma árvore de entidade de relacionamento e aplicação de técnicas de poda e enxerto (não será necessário para o caso de uso deste trabalho), fazer o refinamento das dimensões através de técnicas como o tratamento do relacionamento N por M, tratamento de dimensões heterogêneas, tratamento de Mini-dimensões, removendo atributos desnecessários, estabelecendo hierarquias, tratamento de rastreamento dos atributos da dimensão. Após isto ainda deve ser tratada a Tabela de Fatos, verificando o tipo de fato (linhas de itens ou snapshot), os produtos heterogêneos, classificação dos atributos informando se são chaves, dimensões degeneradas ou atributos numéricos. Depois, deve-se informar o tipo de agregação de cada Fato. Por último, os atributos podem ser nomeados de acordo com os termos do negócio.

- **FASE D – INTEGRAR O DM AO DW :** A partir dos data marts criados nas fases anteriores, é realizada a integração, verificando dimensões comuns, atualizando dimensões, entre outros ajustes.

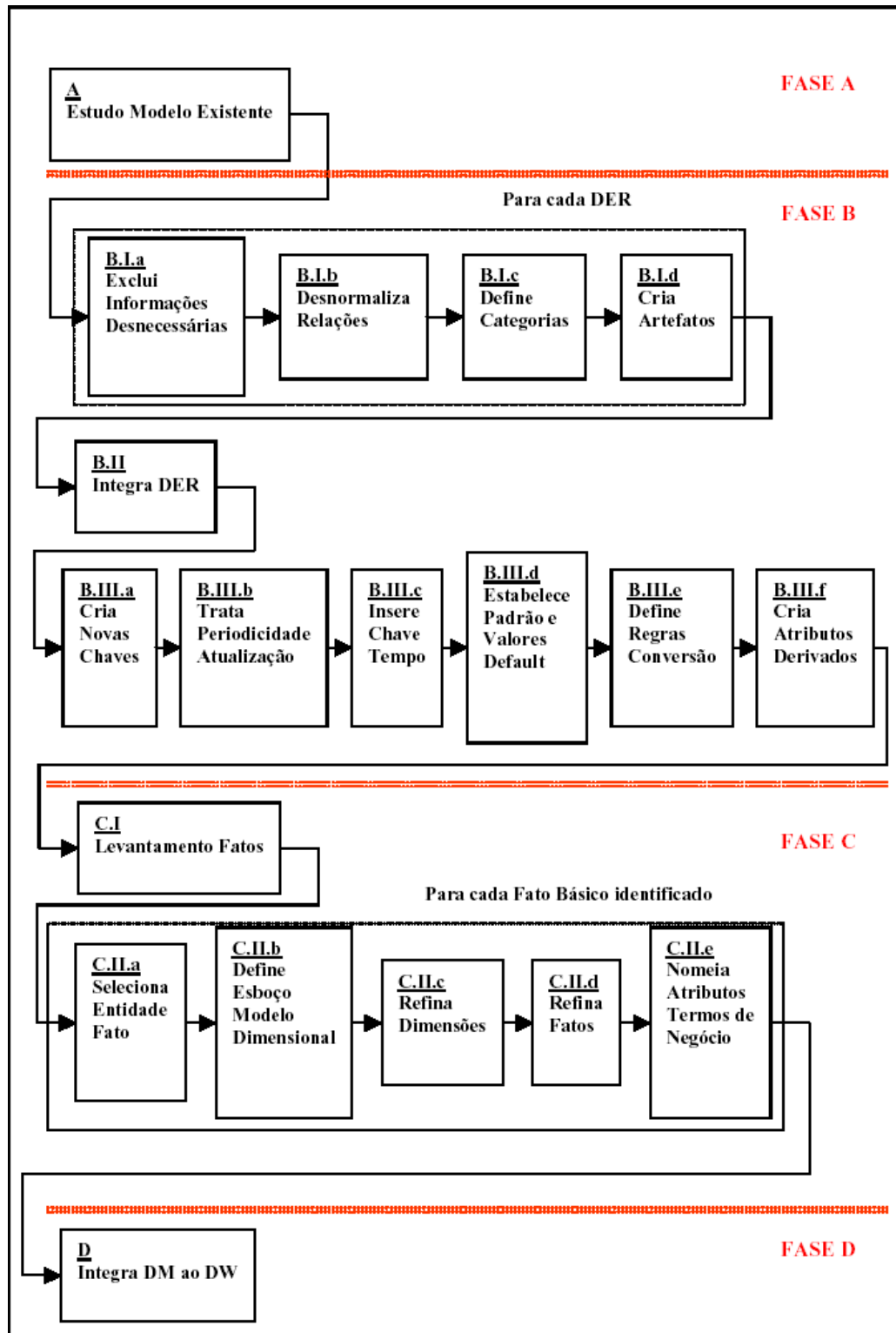


Figura 19: Diretrizes para a Modelagem de um ambiente de DW Incremental
 Fonte: SOARES, 1998 [SOA1998]

CAPÍTULO 4

APLICAÇÃO: SISTEMA DE APOIO A DECISÃO PARA UMA FÁBRICA NACIONAL DE AUTOMÓVEIS

O cenário criado para o estudo de caso deste trabalho é um ambiente propício para a aplicação dos conceitos abordados anteriormente. Este cenário está representado por uma grande empresa nacional do setor automobilístico que deseja desenvolver um ambiente de suporte a decisão para ser usado pelos seus analistas de negócios. Dentre os principais objetivos desta empresa, é possível citar: um maior controle sobre sua produção, um melhor relacionamento com seus clientes e, principalmente, um aumento substancial no seu lucro.

Basicamente, a estrutura desta empresa é formada por uma montadora localizada no Recife e várias concessionárias espalhadas pelo país. Por decisão estratégica dos analistas, primeiramente seria desenvolvido um data mart na montadora, para depois ser construído outro na Central de Concessionárias, localizado em São Paulo, que possui os dados globais de todas as concessionárias. O propósito deste fluxo de desenvolvimento seria a necessidade de uma rápida avaliação da validade dos investimentos nesta tecnologia e o acompanhamento contínuo de sua implementação.

O data mart da montadora, tem como objetivo:

- Acompanhar a produtividade dos seus funcionários;
- Análise da relação de ganhos e custos, para não obter prejuízo e acumular os veículos nas garagens.

O data mart das concessionárias, tem como propósito:

- Análise dos automóveis mais procurados;

- Acompanhamento dos carros mais defeituosos e geração de um relatório para a montadora para que, caso necessário, faça um “recall” aos proprietários destes veículos.

A metodologia de modelagem do Data Warehouse foi baseada nas etapas mencionadas no capítulo anterior, visando a abordagem da construção *Bottom-up* proposta neste trabalho.

4.1 DATA MART MONTADORA

A seguir serão mostradas em detalhes todas as fases da modelagem do data mart da montadora.

FASE A – ESTUDAR OS MODELOS EXISTENTES E EXCLUIR INFORMAÇÕES DESNECESSÁRIAS

O modelo a ser utilizado neste data mart é o DER do Banco de Dados existente no servidor da montadora, como mostrado na figura 20.

Nesta fase, são identificadas as necessidades do sistema e é retirado atributos e identidades que não estejam inseridos neste contexto.

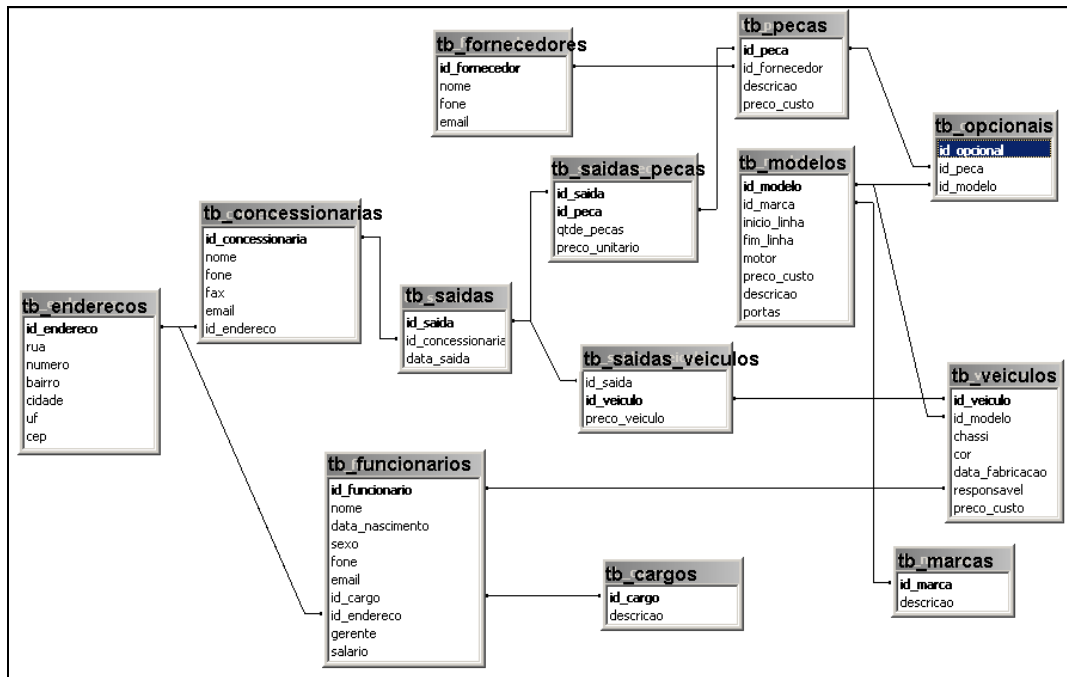


Figura 20: DER Inicial da Montadora

- Objetivo 1: Acompanhar a produtividade dos seus funcionários

Tabelas envolvidas: tb_funcionarios, tb_cargos, tb_veiculos (com seu modelo e marca).

- Objetivo 2: Análise da relação de custos e ganhos (carros vendidos, não vendidos, peças vendidas, não vendidas, salários de funcionários).

Tabelas Envolvidas: tb_funcionarios, tb_cargos, tb_veiculos (com seu modelo e marca), tb_saidas, tb_saidas_veiculos, tb_pecas e tb_saidas_pecas

O DER resultante deve preservar as tabelas envolvidas, retirar as tabelas não envolvidas e as referências das suas chaves nas tabelas restantes. O DER resultante desta fase pode ser representado pela figura 21.

Nesta fase foi realizada uma composição da FASE A (estudar os modelos existentes) com a FASE BI.a. (excluir informações desnecessárias) proposta na tese de Vânia Soares, visando o ganho de velocidade na modelagem, sem acarretar em perda de compreensão.

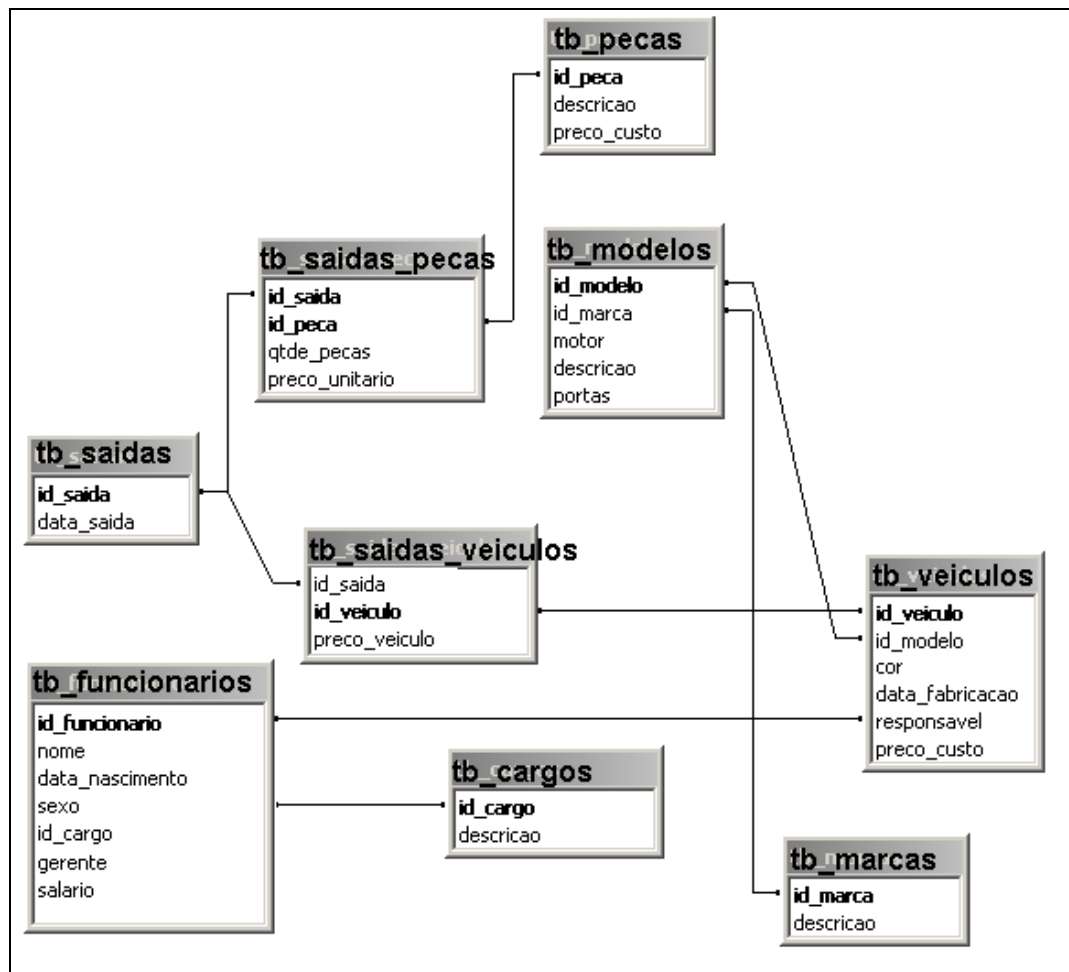


Figura 21: DER Resultante da Fase A e B.I.a.

FASE B.I.b – DESNORMALIZAR RELAÇÕES

Segundo Vânia Soares, esta fase tem o propósito de otimizar as consultas e as futuras extrações de informações para os data marts, através da desnormalizações de entidades.

Verifica-se a possibilidade de desnormalização entre entidade tb_veiculos, tb_marcas e tb_modelos, além de tb_funcionarios – a qual sofre desnormalização com a tb_cargos.

FASE B.I.c – DEFINIR CATEGORIAS

O propósito de se estabelecer categorias é controlar a abrangência que pode ser gerada no momento das consultas. A entidade tb_funcionario possui os seguintes atributos a serem categorizados: data_nascimento e salario. As regras estabelecidas para se definir as categorias para estes campos estão descritas no quadro 7.

Quadro 7: Regras de definição das Categorias

Categoria	Atributo	Regra
Idade	data_nascimento	Cálculo: (data_atual) -(data_nascimento)
faixa_salarial	salario	(salario < 500) = “baixa”; (500 < salário < 2500) = “media”; (salário >= 2500) = “alta”

A próxima fase seria a fase B.I.c (criar artefatos), que consiste basicamente na transformação de entidades em atributos. Por exemplo, se houvesse uma entidade denominada teste_qualidade ligada ao veículo, após a transformação desta fase poderia ser criado um atributo em veículo chamado “testado”, informando se o veículo foi testado ou não. Como não foi identificada a necessidade para este caso de uso, não foi necessário utilizá-la.

Este diagrama também não necessita de integração (Fase B.II – Integrar os DER resultantes)

O DER resultante da fase de limpeza e transformação pode ser representado pela figura 22.

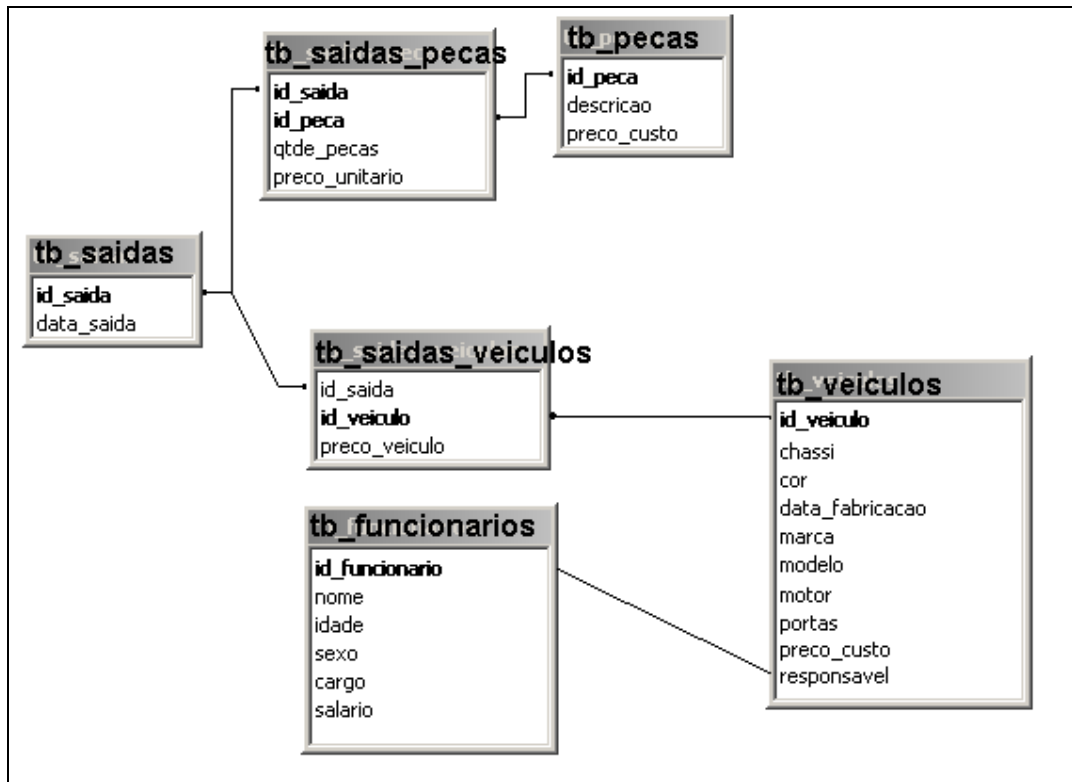


Figura 22: DER Resultante da Fase de Limpeza

A fase B.III.a. cita a necessidade de criar novas chaves para simplificar o acesso. Porém, neste caso não existem chaves longas e esta fase não será empregada. As fases B.III.b até a B.III.d também não são aplicadas a este modelo.

B.III.e. Estabelecer Regras de Conversão para Substituir códigos e abreviaturas

O único atributo a ser derivado apresenta-se descrito no quadro 8 e está presente na entidade tb_veiculos. Nesta derivação, os valores do atributo “portas” que tinham valores definidos como 3 ou 5, são substituídos por “2 portas” ou “4 portas”, respectivamente. Nesta regra de derivação, todos os valores são substituídos por “2 portas” exceto os valores definidos como 5, que serão substituídos por “4 portas” (5 → “4 portas”), como mostra o quadro 8.

Quadro 8: Regras para a Substituição de Códigos e Abreviaturas

CAMPO DERIVADO	DEFAULT	REGRAS DE DERIVAÇÃO
Portas	“2 PORTAS”	5 → “4 PORTAS”

FASE C: ELABORAR O MODELO DIMENSIONAL

FASE COMPOSTA:

C.I. – Realizar o Levantamento de Fatos Básicos e Visões Dimensionais

C.II. – Derivar os Modelos Dimensionais

Através do pré-modelo gerado na fase B, observa-se os seguintes fatos básicos:

- veículos pelo funcionário responsável e pelo modelo;
- funcionários por faixa salarial e por cargo
- saída de veículos por data de saída e pelo modelo;
- saída de peças por data de saída.

Através destes fatos básicos é possível realizar as seguintes observações:

- a) Para acompanhar a produtividade dos funcionários pode-se analisar os veículos pelo responsável, além de observar os funcionários de acordo com uma certa faixa salarial ou cargo.
- b) Para fazer a análise dos ganhos e gastos da empresa, é possível acompanhar a saída de peças de acordo com a data e a saída de veículos, pela data e modelo.

Os modelos dimensionais são derivados a partir de cada fato básico e do pré-modelo gerado na fase B. Como o foco deste capítulo é mostrar a maneira como é feita modelagem incremental e não gerar um fluxo longo e pouco prático deste processo de desenvolvimento, o trabalho será concentrado apenas no primeiro fato básico e sua posterior integração ao Data Warehouse.

- FATO BÁSICO: veículos pelo funcionário responsável e pelo modelo

C.II.a. Selecionar Entidade Chave Relacionada ao Fato Básico

Este fato básico utiliza a entidade tb_veiculos. A tabela de fatos que será criada será denominada “producao”. Como este fato é bastante simples, não será necessária a aplicação das fases C.II.b.i. e C.II.b.ii.

C.II.b.iii. Criar a dimensão tempo

Como os veículos são produzidos diariamente, a granularidade será o dia.

C.II.c.iii. Elaborar Minidimensões

A minidimensão é constituída de informações relacionadas ao fato. A sua utilização permite um melhor desempenho das consultas e uma melhor visualização da dimensão. Na tabela de funcionários pode ser extraída a minidimensão demográfica (sexo, salário e idade).

C.II.c.iv. Estabelecer Hierarquias

Para a dimensão tempo, pode ser estabelecida a seguinte hierarquia: Ano, Mês e Dia.

O próximo passo seria a etapa C.II.d.i. (Tratar Atributos de Acordo com Tipo de Fato), porém o fato “producao” já se apresenta na granularidade desejada, não sendo necessário nenhum tratamento de atributo.

C.II.d.iii. Classificar os Atributos da Tabela de Fato

Podem ser estabelecidos os seguintes atributos para a entidade producao:

- Chaves de dimensões: id_tempo, id_funcionario, id_veiculo e id_demografica.
- Valor semi-aditivo: preco_custo

O modelo dimensional para o fato producao está representado na figura 23.

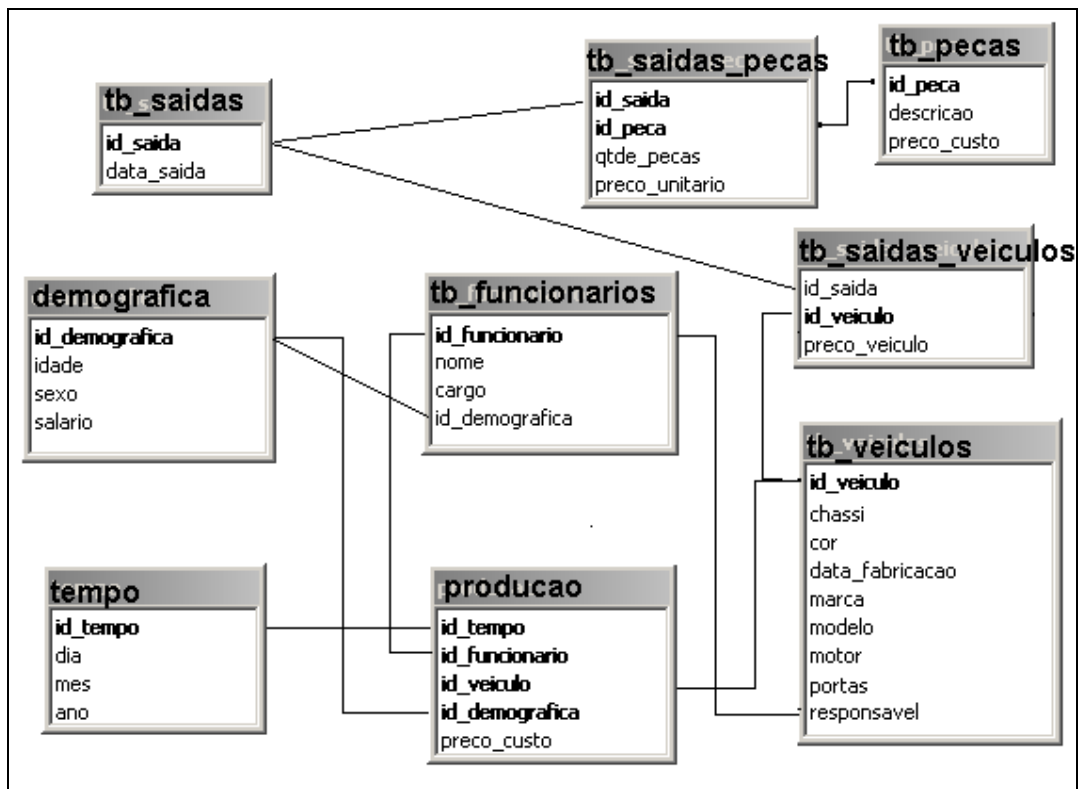


Figura 23: Data Mart Montadora

FASE D – INTEGRAR O DATA MART AO DATA WAREHOUSE

Na metodologia incremental proposta por Kimball [KIM1998], o data mart final deve ser formado pela união dos modelos dimensionais (no caso exemplo, o data mart seria o próprio modelo da figura 24). Após concluir a modelagem do data mart, deve-se realizar a integração com o Data Warehouse (DW). Como este foi o primeiro data mart a ser modelado, o próprio representa o DW.

4.2 DATA MART CONCESSIONÁRIA

O próximo passo agora é construir o data mart da concessionária e posteriormente integrá-lo ao DW. A seguir serão mostradas as fases de modelagem para este novo departamento.

FASE COMPOSTA:**FASE A – ESTUDAR OS MODELOS EXISTENTES****FASE B1.a. – EXCLUIR INFORMAÇÕES DESNECESSÁRIAS**

O modelo a ser utilizado neste data mart é o DER do Banco de Dados existente no servidor da montadora, como mostrado na figura 24.

Como dito anteriormente, nesta fase são identificadas as necessidades do sistema e são retirados atributos e identidades que não estejam inseridos neste contexto.

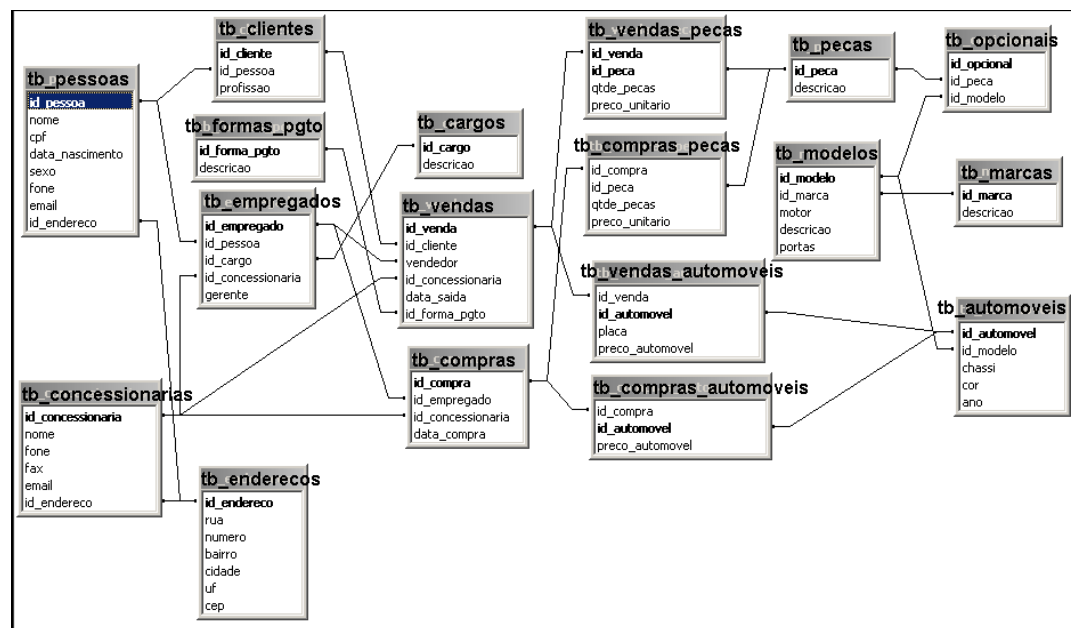


Figura 24: DER Concessionária

Como principais objetivos da implementação de um data mart na rede de concessionárias, apresentam-se:

- Objetivo 1: Análise detalhada dos automóveis mais procurados, visando o aperfeiçoamento dos mesmos e a satisfação dos clientes.

Tabelas envolvidas: tb_pessoas, tb_clientes, tb_vendas, tb_vendas_automoveis, tb_concessionarias, tb_enderecos, tb_automoveis, tb_marcas e tb_modelos.

- Objetivo 2: Acompanhamento dos carros mais defeituosos e geração de um relatório para a montadora para que, caso necessário, faça um “recall” aos proprietários destes veículos

Tabelas Envolvidas: tb_automoveis, tb_modelos, tb_marcas, tb_pecas, tb_compras_pecas, tb_compras, tb_concessionarias, tb_enderecos

Em um caso real de uma rede de concessionárias, existem muitos outros objetivos para satisfazer as necessidades dos tomadores de decisões. Porém, para atender o escopo prático deste estudo, os outros objetivos, assim como o objetivo 2, não serão explorados nesta seção. Para a construção do DER resultante após a eliminação das tabelas não envolvidas será considerado apenas o objetivo 1. Desta forma, o DER resultante desta fase pode ser representado pela figura 25.

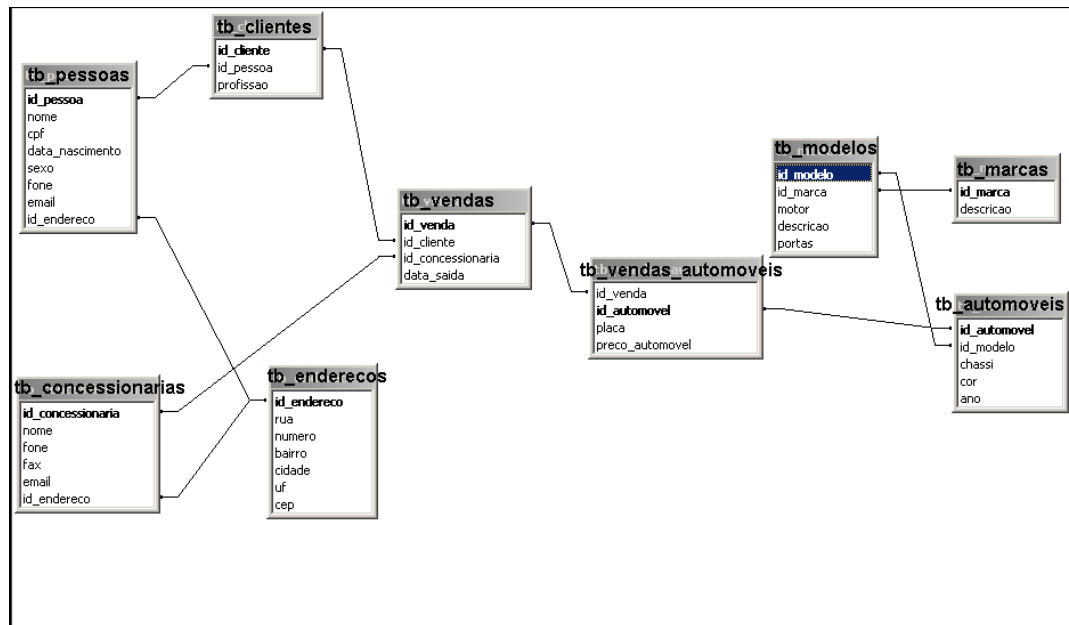


Figura 25: DER Resultante da fase B.I.a

FASE B.I.b – DESNORMALIZAR RELAÇÕES

Verifica-se a possibilidade de desnormalização entre as seguintes entidades: tb_clientes com tb_pessoas e tb_enderecos; tb_concessionarias com tb_enderecos; tb_automoveis com tb_modelos e tb_marcas.

FASE B.I.c. – DEFINIR CATEGORIAS

O único atributo que será categorizado será a data_nascimento em tb_pessoas. A regra estabelecida para se definir a categoria idade é a mesma definida para o data mart montadora.

Neste data mart também não será necessária a fase B.I.c (criar artefatos), ou seja, a transformação de entidades em atributos, nem a integração com outros DER resultantes (Fase B.II)

O DER resultante da fase de limpeza e transformação pode ser representado pela figura 26.

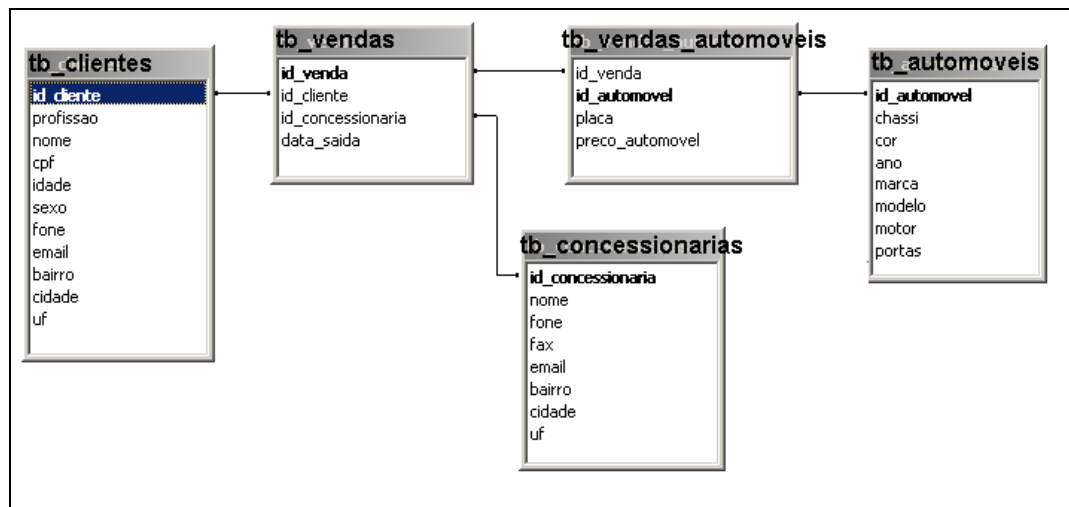


Figura 26: DER Resultante da Fase de Limpeza

As fases B.III.a. até a B.III.d também não são aplicadas a este modelo pelos mesmos motivos mencionados no data mart anterior.

B.III.e. Estabelecer Regras de Conversão para Substituir códigos e abreviaturas

Os atributos a serem derivados apresentam-se descritos no quadro 9 e estão presentes na entidade tb_veiculos e tb_clientes, respectivamente.

Quadro 9: Regras para a Substituição de Códigos e Abreviaturas

CAMPO DERIVADO	DEFAULT	REGRA DE DERIVAÇÃO
Portas	“2 PORTAS”	5 -> “4 PORTAS”
Sexo	“M”	“H” -> “M” “M” -> “F”

FASE C: ELABORAR O MODELO DIMENSIONAL

FASE COMPOSTA:

C.I. – Realizar o Levantamento de Fatos Básicos e Visões Dimensionais

C.II. – Derivar os Modelos Dimensionais

Através do pré-modelo gerado na fase B, observa-se o seguinte fato básico:

- vendas por clientes, concessionárias, por automóveis, por estados ou por período

O modelo dimensional será realizado a partir do fato básico proposto e o do pré-modelo gerado na fase B. Como não foi gerado nenhum outro fato básico, não será realizado a integração dos modelos dimensionais antes da integração deste data mart ao Data Warehouse.

- FATO BÁSICO: vendas por clientes, concessionárias, por automóveis, por estados ou por período

C.II.a. Selecionar Entidade Chave Relacionada ao Fato Básico

Este fato básico utiliza a entidade tb_vendas_automoveis. A tabela de fatos que será criada será denominada “mercado”. Como este fato é bastante simples, não será necessária a aplicação das fases C.II.b.i. e C.II.b.ii.

C.II.b.iii. Criar a dimensão tempo

Como os veículos são vendidos diariamente, a granularidade será o dia.

C.II.c.iii. Elaborar Minidimensões

Nas tabelas `tb_clientes` e `tb_concessionarias` pode ser extraída a minidimensão espacial (estado, cidade e bairro) e demográfica (sexo, profissão e idade), para `tb_clientes`.

C.II.c.iv. Estabelecer Hierarquias

Para a dimensão tempo, pode ser estabelecida a seguinte hierarquia: Ano, Mês e Dia. Também pode ser estabelecida a seguinte hierarquia na dimensão espaço: Estado, Cidade e Bairro.

O próximo passo seria a etapa C.II.d.i. (Tratar Atributos de Acordo com Tipo de Fato), porém o fato “mercado” já se apresenta na granularidade desejada, não sendo necessário nenhum tratamento de atributo.

C.II.d.iii. Classificar os Atributos da Tabela de Fato

Podem ser estabelecidos os seguintes atributos para a entidade mercado:

- Chaves de dimensões: `id_tempo`, `id_venda`, `id_automovel`, `id_veiculo`, `id_concessionaria`, `id_cliente` e `id_demografica`.
- Valor semi-aditivo: `preco_automovel`
- Valor não-aditivo: `placa`

O modelo dimensional para o fato mercado está representado na figura 27.

FASE D – INTEGRAR O DATA MART AO DATA WAREHOUSE

Comparando o modelo do data mart concessionaria e o DW (figuras 28 e 24, respectivamente) verifica-se a existência de duas entidades com nomes diferentes, porém as informações são praticamente as mesmas: `tb_veiculos` e `tb_automoveis`. Por esta razão, deve ser adequado o nome da `tb_automoveis` ao Data Warehouse existente. Nesta mesma tabela, percebe-se que não existe o atributo “responsavel”. Portanto, na tabela `tb_automoveis`, também será criado um atributo “responsavel”, com valor *default* nulo. Por conveniência, a entidade `tb_vendas_automoveis` será renomeada para `tb_vendas_veiculos`, o campo `preco_automovel` será denominado `preco_veiculo` e a chave `id_automovel` presente nas tabelas `tb_vendas_automoveis` e `mercado` (tabela fato) será renomeada para `id_veiculo`.

As entidades “demografica” e “tempo” estão presentes tanto no data mart quanto no Data Warehouse atual. Por isso, não precisam ser integradas. As outras tabelas presentes no data mart não existem no DW e, como não causam conflitos, serão adicionadas ao modelo existente.

O Data Warehouse final, obtido pelo processo incremental de integração, pode ser representado através da figura 28.

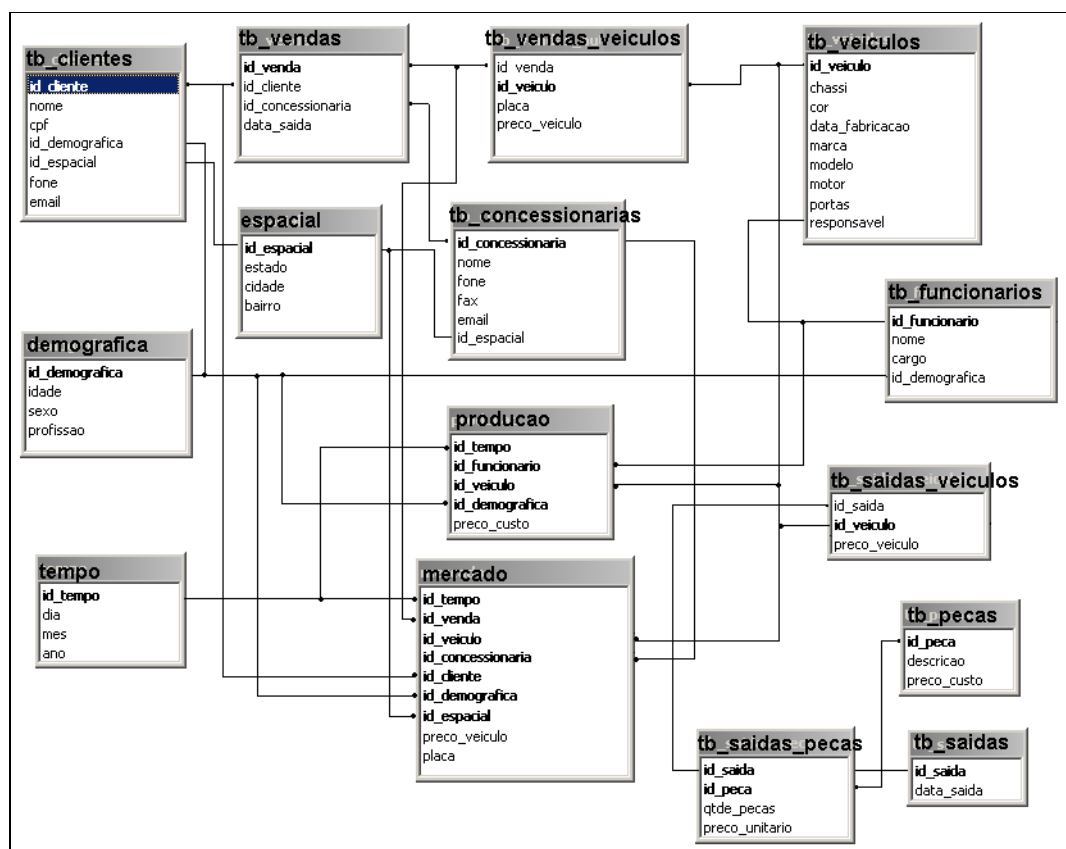


Figura 28: Data Warehouse Global

4.3 ANÁLISE DOS RESULTADOS

Logo após a construção do primeiro data mart através das etapas descritas anteriormente e uma suposta carga posterior, o tomador de decisões já pode ser beneficiado com as consultas analíticas aos dados. Como exemplo, após o desenvolvimento do data mart Concessionária, os analistas de negócios podem

examinar os variados aspectos das vendas por concessionárias, estados e por período. No anexo B seguem as telas de análises que poderiam ser realizadas por um funcionário de alta chefia dentro desta suposta organização através da utilização de uma ferramenta OLAP de código aberto (*open source*) chamada Jpivot, que pode ser disponibilizada na *Web*. As figuras deste anexo retratam a definição do nível de detalhe que esta ferramenta pode alcançar.

CAPÍTULO 5

CONCLUSÃO E TRABALHOS FUTUROS

O objetivo principal deste trabalho é o de realizar uma análise de uma arquitetura em conjunto com técnicas de modelagem para distribuição de um Data Warehouse que proporcione o acompanhamento contínuo do tomador de decisões. Desta forma, foram divididos nos capítulos deste trabalho, os elementos teóricos necessários, a discussão sobre a organização da arquitetura e modelagem e sua implementação.

Um Data Warehouse é indicado quando o projeto de um sistema de informações necessita agregar dados de vários sistemas operacionais. Em diversas situações, a performance das informações obtidas a partir dos sistemas operacionais pode ser comprometida, devido a fatores, tais como, diversidade de plataformas de sistemas operacionais e sistemas gerenciadores de banco de dados, informações duplicadas, representações diferentes dos mesmos dados, entre outras.

Além disto, a amplitude temporal (informações de meses e anos anteriores) e os dados externos desejados pelos usuários podem não estar disponíveis nos sistemas operacionais, visto que tais sistemas foram planejados para atender ao processamento das transações do dia a dia, e não para fins analíticos.

Na implementação de um Data Warehouse, a escolha da metodologia de desenvolvimento e a definição da arquitetura de implementação, constituem tarefa de suma importância para o sucesso do projeto. Dentre as arquiteturas apresentadas no terceiro capítulo, aponta-se como mais adequada a baseada em data marts incrementais, denominada *Bottom-up*, por reunir vantagens das metodologias baseadas em Data Warehouse global e data marts isolados.

A arquitetura *Bottom-up* foi proposta para aplicação em conjunto com a metodologia de data marts incrementais na intenção de proporcionar máxima

flexibilidade de adaptação a mudanças, fator crucial para o sucesso de um projeto de Data Warehouse.

A eficácia da arquitetura foi comprovada quando da criação de um cenário operacional de uma empresa automobilística e, em seguida, o desenvolvimento de um sistema de informações, atendendo assim, a suposta demanda de informações dos seus analistas. A utilização da arquitetura permitiu que várias modificações oriundas de mudanças nos requisitos dos usuários fossem implementadas de forma rápida, sem comprometer o funcionamento do sistema.

Após o estudo do caso proposto, verificou-se que poucas modificações nas fases de modelagem apresentadas na tese de Vânia Soares [SOA1998] foram sugeridas. Para um estudo mais aprofundado, a aplicação de um caso real é fortemente indicada para avaliar as etapas referentes à modelagem e propor mudanças. Apesar disto, este trabalho mostrou-se extremamente válido para a complementação literária das metodologias focadas na arquitetura *Bottom-up*, visando efetivá-las através de diversas aplicações em casos reais, favorecendo a melhoria do conhecimento voltado para a área de Data Warehouse e, conseqüentemente, Banco de Dados.

5.1 TRABALHOS FUTUROS

Como seqüência deste trabalho, são identificados os seguintes trabalhos que permitem especificar e implementar sistemas nesta área:

- Um sistema especialista para gerenciamento do metadados. Com a utilização de um sistema especialista, com regras definidas para a extração e apresentação dos dados, a manutenção do metadados se tornaria mais fácil e haveria uma maior flexibilidade para adaptações às mudanças, como, por exemplo, a indicação dos data marts afetados pela modificação de uma dimensão;
- Especificação de interfaces para cada módulo da área de extração, para sugerir um modelo de especificação padrão para componentes da área de extração do Data Warehouse;

- Construção de ferramentas de extração de dados do ambiente operacional para o Data Warehouse ou data mart. Nesta ferramenta, seriam geradas as consultas e os conseqüentes resultados a serem armazenados no DW, a partir da indicação dos campos modificados pela modelagem destes ambientes analíticos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ARM2005] ARMSTRONG, R., *Data Warehousing: Clearing the Confusion*. Disponível em: <<http://www.ncr.com>>. Acesso em 10 jan. 2005.
- [BER1997] BERSON, A., *Data Warehousing, Data Mining & OLAP*. McGraw-Hill, USA, 1997.
- [CHU1998] CHUCK, B., DIRK, H., DON, S., et al. *Data Modeling Techniques for Data Warehousing*. 1998.
- [COR1998] COREY, Michael J., ABBEY, Michael. *Oracle Data Warehousing*. Califórnia: Oracle Press™ Edition, 1996.
- [DEB1999] DEBEVOISE, N. T., *The Data Warehouse Method*. Prentice Hall, New Jersey, 1999.
- [DON1997] DONALD, B., *High Performance Oracle Data Warehousing*. The Coriolis Group, USA, 1997.
- [GIO2000] GIOVINAZZO, W. A., *Object-Oriented Data Warehouse Design - Building a Star Schema*. Prentice Hall, New Jersey, 2000.
- [HAC1998] HACKNEY, Douglas. *Data Warehouse Delivery: Who Are You? – Part I*. DM Review Magazine, Vol. 8, No 2, Fevereiro, 1998.
- [HAR1997] HARRISON, T. H., *Intranet Data Warehouse*. McGraw-Hill, USA, 1997.
- [INM1997] INMOM, *Como construir o Data Warehouse*. Rio de Janeiro: Editora Campus, 1997.
- [KIM1998] KIMBALL, Ralph. *Data Warehouse Toolkit*. São Paulo: Makron Books, 1998.
- [LAU1999] LAUDON, K. C. & LAUDON, J. P., *Management Information Systems. New Approaches to Organization & Technology*. Prentice Hall, New Jersey, 1998.

- [MAC2000] MACHADO, Felipe Nery Rodrigues. *Projeto de Data Warehouse – Uma visão Multidimensional*. São Paulo: Érica, 2000.
- [MEL1997] MELO, Rubens Nascimento. *Data Warehousing (Tutorial)*. In: XIII SBBB – Simpósio Brasileiro de Banco de Dados, Salvador, 1997.
- [MEY1998] MEYER, Don, CANNON, Casey. *Building a Better Data Warehouse*. New Jersey: Prentice-Hall PTR, 1998. 227 p.
- [NEI1997] NEIL, B., SCHRADER, M., DAKIN, J., *Oracle Data Warehousing Unleashed*. Publishing, Inc., 1997.
- [PAL1998] PALMA, S., *Os Componentes Funcionais de um Data Warehouse*. In: *Developers' Magazine*, Rio de Janeiro: Axcel Books, 1998.
- [POE1998] POE, Vidette, KLAUER, Patricia, BROBST, Stephen. *Building a Data Warehouse for Decision Support*. New Jersey. Prentice-Hall, Inc, 1998. 285 p.
- [SAN2005] SANTOS, R. Ângelo F., *Fastcube: Uma Metodologia de Construção de Data Mart*. Disponível em: <http://infodw.port5.com/aula_dm.pdf>. Acesso em 07 jan. 2005.
- [SIL1997] SILVERSTON, Len, INMON, W.H., GRAZIANO, Kent. *The Data Model Resource Book - A Library of Logical Data Models and Data Warehouse Designs*. Nova York: Wiley Computer Publishing, 1997. 335 p.
- [SIN2001] SINGH, Harry S. *Data Warehouse: Conceitos, Tecnologias, Implementação e Gerenciamento*. São Paulo: Makron Books, 2001.
- [SOA1998] SOARES, Vânia. *Modelagem Incremental no Ambiente de Data Warehouse*. Rio de Janeiro: UFRJ/IM/NCE, 1998 216 p.
- [TAN1998] TANLER, Rick. *Intranet Data Warehouse*. Rio de Janeiro: Livraria e Editora Infobook S.A., 1998. 394 p.
- [THO1997] THOMSEN, Erik. *OLAP Solutions - Building Multidimensional Information Systems*. New York: John Wiley & Sons, Inc , 1997. 576 p.

- [VAS1999] VASCONCELOS, J. M., *Implementando um Data Warehouse Incremental*. In: *Developers' Magazine*, Rio de Janeiro: Axcel Books, 1999.
- [UNI2005] UNIVERSITY OF ILLINOIS, *Desicion Support - Information Data Access*. Disponível em <<http://www.ds.uillinois.edu>> Acesso em: 20 fev. 2005.

ANEXO A

Quadro 10: Papéis em um Projeto de Data Warehouse

Papel	Descrição
Gerente de Projeto	Administração das tarefas, atividades e recursos, gerência dos prazos e comunicação do progresso do projeto
Analista da Aplicação cliente	Definição dos requisitos e representação destes na forma de um modelo dimensional a ser desenvolvido
Modelador de Dados	Análise detalhada e construção do modelo dimensional de dados. Participação secundária na análise de requisitos
Analista de Normas	Buscar definições comuns na organização para dimensões e fatos, a fim de publicar tais definições
Administrador de Banco de Dados (DBA)	Traduzir o modelo dimensional em tabelas, definir estratégias de indexação, particionamento, integridade dos dados e tarefas ligadas a manutenção a fim de manter o banco de dados disponível para uso
Projetista da Extração dos Dados	Projeto dos processos de extração, transformação, carga dos dados dos sistemas operacionais no Data Warehouse
Desenvolvedor da Extração dos Dados	Desenvolvimento dos programas de extração em caso de não adoção de ferramentas automatizadas
Desenvolvedor da Aplicação cliente	Desenvolvimento da aplicação que será disponibilizada ao usuário final
Educador do Data Warehouse	Treinamento sobre as aplicações dos usuários e conteúdo dos dados
Analista de Controle de Qualidade	Verifica o cuidado no carregamento dos dados no Data Warehouse e a integridade da aplicação cliente

Fonte: KIMBALL, 1998 [KIM1998]

ANEXO B

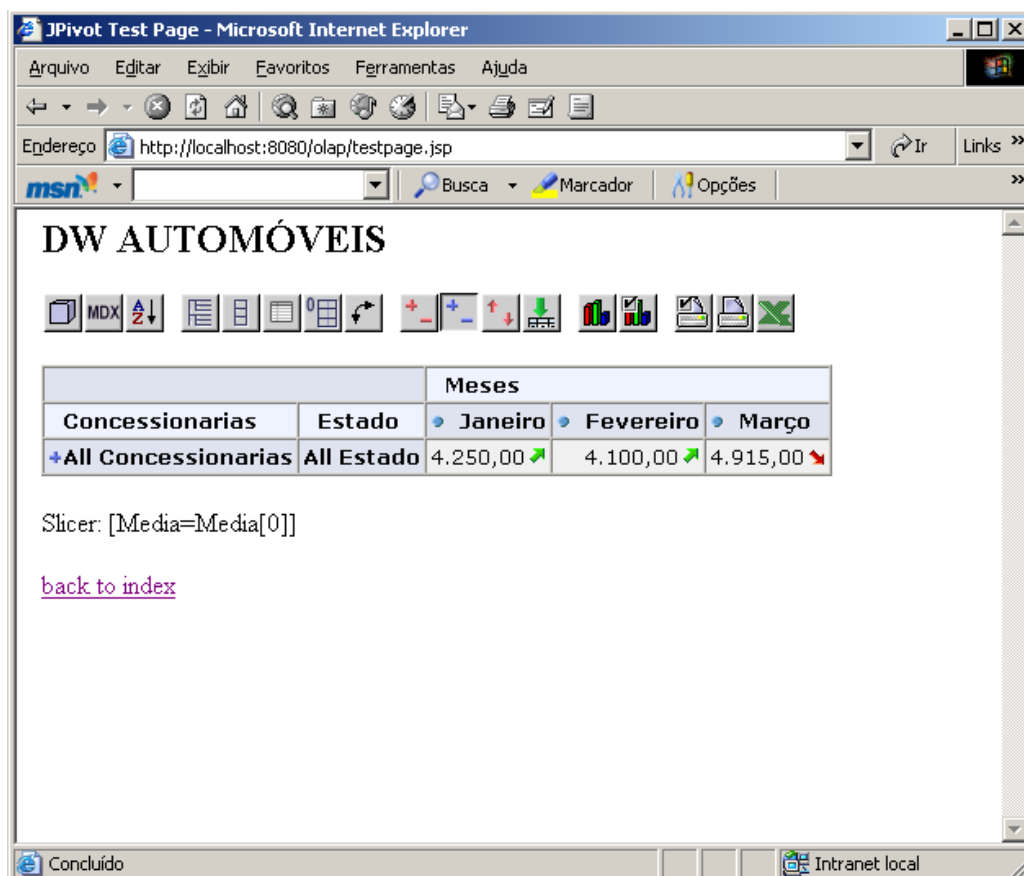


Figura 29: Valores distribuídos no tempo e agrupados por concessionárias e estados.

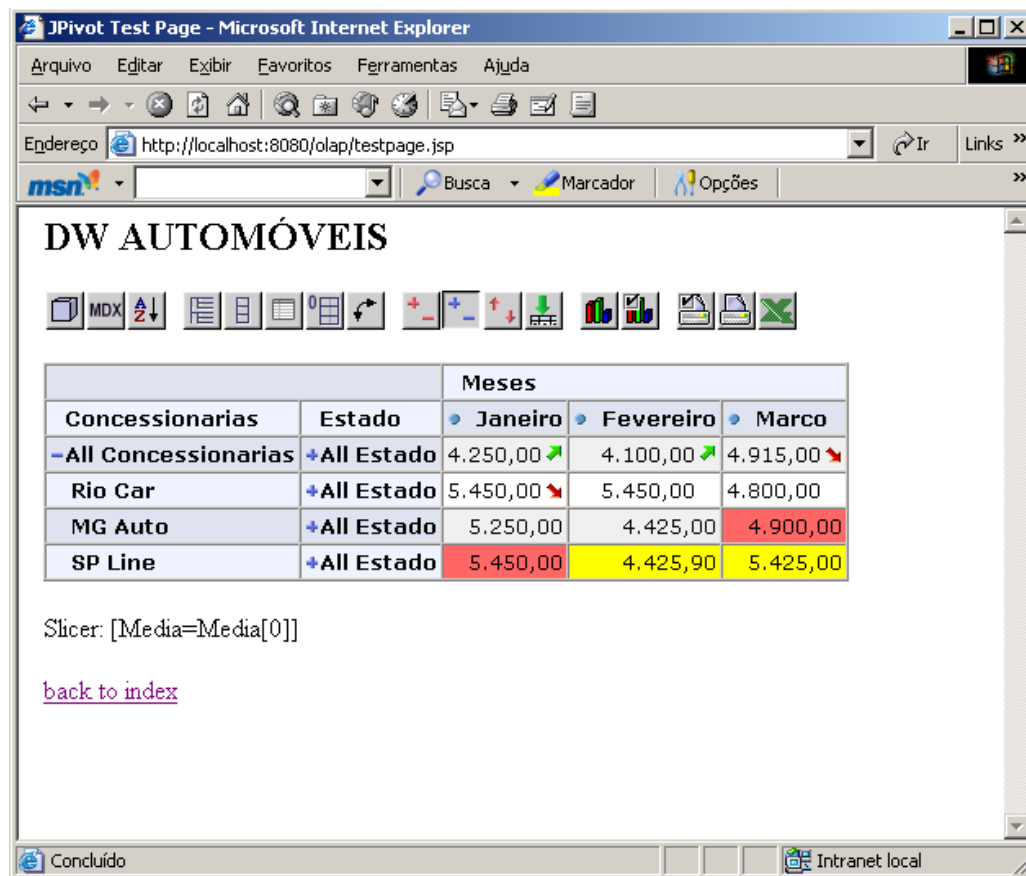


Figura 30: *Drill-down* nas concessionárias

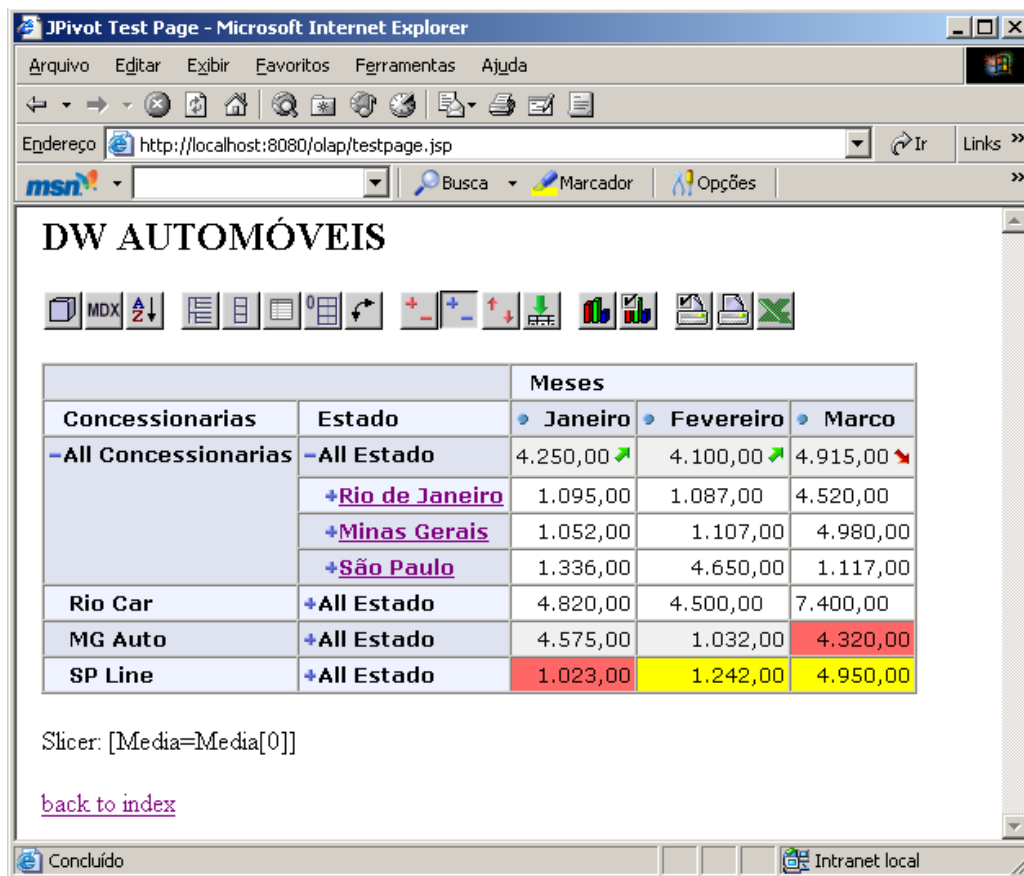


Figura 31: Drill-down nos estados

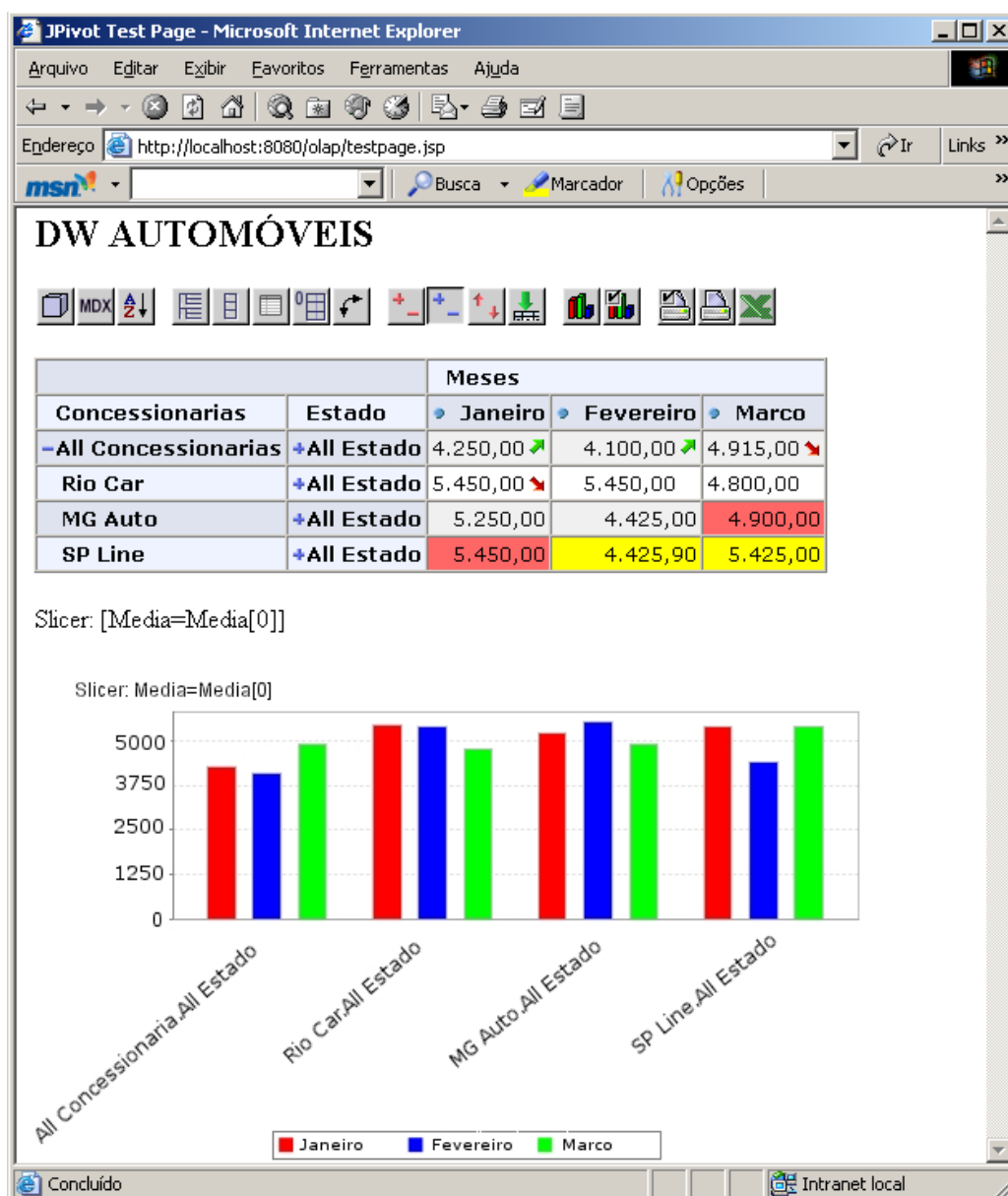


Figura 32: Representação gráfica dos valores por concessionária e por Estado