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

LAURA COSTA SARKIS

DATA WAREHOUSE: O PROCESSO DE MIGRAÇÃO DE DADOS.

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Murilo Silva de Camargo

Florianópolis, Fevereiro 2001.

DATA WAREHOUSE: O PROCESSO DE MIGRAÇÃO DE DADOS

LAURA COSTA SARKIS

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

_____
Fernando Ostuni Gauthier. Dr.
Coordenador do PPGCC da UFSC

Banca Examinadora

_____
Murilo Silva de Camargo , Dr. (Orientador)
Universidade Federal de Santa Catarina _UFSC

_____
Darlene F. B. Coelho, Drª.
Universidade Federal de Rondônia - UNIR

_____
Fábio Paraguaçu Duarte da Costa , Dr.
Universidade Federal de Alagoas UFAL

_____
Luiz F. J. Maia, Dr.
Universidade Federal de Santa Catarina - UFSC

A Deus, que possibilitou a realização deste trabalho;

A minha avó, Aos meus pais (Sérgio e Francisca);

Aos meus irmãos e sobrinhos;

A meu esposo Jorge e meu filho Matheus, meus grandes companheiros e motivadores;

E a meu filho Nícolas – mais uma pérola em minha vida que está para chegar.

Agradecimentos

À Universidade Federal do Acre – UFAC, pela oportunidade;

Aos professores do Departamento de Matemática e Estatística da UFAC que me compreenderam, incentivaram e foram companheiros nas horas que precisei;

Ao convênio UNIR/UFSC, sem o qual não seria possível a realização deste mestrado;

A Murilo da Silva Camargo, orientador deste trabalho;

A todos que me deram muito apoio, incentivo, e aos que diretamente ou indiretamente possibilitaram a realização deste trabalho.

E especialmente a Jorge e Matheus que não mediram esforços e sacrifícios para estarem presentes em todos os momentos, por suportarem as minhas ausências - mesmo estando presente, pela compreensão, pelo amor, carinho e pelo total incentivo dado a mim durante esta jornada.

Sumário

Índice de figuras

Índice de Tabelas

Resumo

Este trabalho descreve os conceitos básicos do ambiente do *Data Warehouse*, abordando em especial o processo de migração de dados. São expostas algumas técnicas e tecnologias mais recentes existentes no mercado com esta finalidade. A partir de um estudo inicial sobre os conceitos de *Data Warehouse*, delimitou-se o trabalho em função do processo de migração dos dados. Com este propósito, foram estudadas quatro abordagens e elaborada uma análise comparativa na tentativa de determinar qual delas é a mais adequada ao processo. Em um processo de migração de dados é importante garantir também a qualidade dos dados, em decorrência disto, o trabalho contém a descrição de uma abordagem que trata de como é realizado o processo para a qualidade de dados em *Data Warehouse*. São citadas também algumas ferramentas existentes no mercado que possam possivelmente atender aos processos de migração de dados para o *Data Warehouse* e qualidade de dados.

Palavras-chaves: *Data Warehouse*, Migração de Dados e Qualidade de dados.

Abstract

This work describes the basic concepts of *Data Warehouse* environment, especially approaching the data migration process. They are exposed some techniques and existent more recent technologies in the market with this purpose. Starting from an initial study about *Data Warehouse* basic concepts, this work was delimitated in function of data migration process. With this purpose, four approaches were studied and a comparative analysis was made aiming to determine which of them is the most adequate for this process. As important in a data migration process is the data quality assurance. This work includes the description of an approach that handles how data quality process is made in *Data Warehouse*. Some tools existent in the market that may provide data migration and data quality processes in *Data Warehouse* are mentioned.

Keywords: *Data Warehouse*, Data Migration and Data Quality.

CAPÍTULO I

Introdução

1.1. Introdução

A disponibilidade de metodologias, bem como, o acesso aos dados objetivando a busca de informações, é de grande importância, principalmente no mundo dos negócios, onde pode-se ter o ponto de diferenciação de empresas bem sucedidas.

Atualmente no mercado de negócios, já existem ferramentas e técnicas aplicáveis para atender anseios das corporações, tais como: identificar padrões e tendências de negócios para impulsioná-las no mercado. Já é possível por exemplo, estabelecer técnicas para disponibilizar produtos em prateleiras de supermercados, objetivando aumento nas vendas. Mas para que ferramentas ou técnicas funcionem corretamente é necessário que os dados sobre os quais se está atuando, sejam dados corretos e completos; caso contrário, as informações geradas a partir destes, não serão informações satisfatórias e seguras.

Com a finalidade de oferecer às organizações um sistema de banco de dados com estes padrões, gerando informações decisivas e de qualidade, vem-se há décadas pesquisando tecnologias. Uma que tem despontado é o *Data Warehouse*, que é um ambiente analítico, capaz de gerar informações para o apoio a decisões gerenciais, através de identificação de padrões e tendências de mercado no negócio da empresa.

Para alguns pesquisadores do assunto, dentre eles [KIM98L] e [GRA98], a arquitetura que envolve o *Data Warehouse*, compõe-se de uma estrutura que trata do *Back-End*, do repositório de dados – *Data Warehouse*, e do *Front-End.*

Para melhor ilustrar a estrutura supracitada, foi extraída de [GRA98] a figura 1, que é uma arquitetura simplificada do *Data Warehouse*:

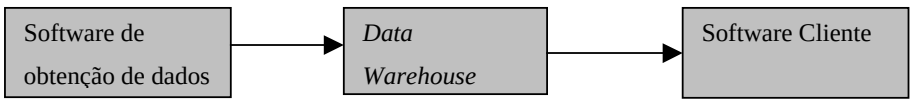

FIGURA 1 - ARQUITETURA SIMPLIFICADA DO DATA WAREHOUSE

Na arquitetura da figura 1, inicialmente os dados são extraídos das fontes, que podem ser sistemas operacionais ou fontes externas, depois passam por transformações (por exemplo, consolidações e sumarizações), para então serem carregados no *Data Warehouse*, que é um banco de dados analítico para suporte à decisão. Uma vez os dados estando armazenados no *Data Warehouse*, os usuários podem acessá-los através de ferramentas de *front-end*, para permitir tomada de decisões.

O *Back-End* é responsável por consumir de 60% a 75% do tempo gasto em projetos de *Data Warehouse*. Isto ocorre porque é neste nível que está um dos processos da migração de dados - a limpeza de dados - uma das partes mais exigentes do processo de *Data Warehousing*.

No processo de Data Warehousing; após a *extração dos dados* dos sistemas fontes, utiliza-se um ambiente intermediário para que a *limpeza dos dados* seja feita – a *Data Staging Área* – Área de estágio de dados. Neste ambiente são identificados os dados sujos provenientes dos sistemas legados, e especificadas regras de identificação do tipo de sujeira. Estas regras são documentadas como *metadados*. Posteriormente, estes *metadados* são consultados com a finalidade de efetivar a limpeza dos dados; para então finalmente, a última atividade de migração dos dados ser executada – a *carga dos dados* no *warehouse*.

Limpar dados não é somente atualizar um registro com dados bons; envolve muito mais. Em uma séria limpeza de dados se faz necessário utilizar decomposição e reagrupamento de dados, tarefas árduas, principalmente se executadas manualmente [KIM996].

Os dados para povoarem o *Data Warehouse* devem ser submetidos aos processos supramencionados, que os tornam aptos às consultas dos analistas, que utilizam estes dados para ajudar a conduzir a empresa nos negócios.

1.2. Justificativa

Nos últimos anos, tem sido imprescindível para as organizações, a tomada de decisões rápidas e precisas, a fim de torná-las muito mais competitivas e rentáveis. Para tal, estas estão sempre se adequando as modificações que ocorrem em seu meio ambiente; buscando sempre ter informações corretas e facilmente acessíveis, para tomada de decisões em um tempo bastante hábil. Com este objetivo, desde a década de 70, um conjunto de tecnologias vem sendo desenvolvido, dentre elas, os Sistemas de Apoio a Decisão (SAD), que através de uma evolução natural proporcionou o surgimento do *Data Warehouse*(DW). Estes sistemas analíticos, servem para os usuários identificarem de forma geral as ações e modelos que devem ser seguidos pela organização a qual pertencem, tornando a tarefa de acesso a informação mais dinâmica, rápida e confiável; uma vez que as informações encontram-se mais compactas.

O armazenamento de informações através de sistemas transacionais ocorre há décadas nas corporações, assim como, a busca por sistemas computacionais mais eficientes, rápidos e com baixos custos. Desta forma, a busca sempre foi por tecnologias capazes de melhorar as já existentes. Por exemplo: embora seja o ideal começar armazenamento em repositórios de dados totalmente do zero; dos pontos de vista financeiro, humano e do fator tempo, seria muito dispendioso para qualquer corporação que já armazena informações há anos em banco de dados existentes, conceber isto. Neste contexto, a aplicabilidade de técnicas do processo de migração de dados torna-se imprescindível, uma vez que, estas possuem a tarefa de fazer com que dados que estão armazenados em bancos de dados já existentes, sejam capturados (extraídos), transformados e carregados no *Data Warehouse*.

Estudar as técnicas de migração de dados é primordial para construção de um *Data Warehouse*, uma vez que, mais da metade do tempo e pessoal envolvido no projeto de um *Data Warehouse* é destinado a esta fase do projeto. Nesse sentido, este trabalho foi elaborado, a fim de estudar algumas técnicas de migração de dados para o *Data Warehouse*, proporcionando uma visão acadêmica sobre o tema. Pois, embora existam abordagens de migração de dados para o *Data Warehouse*, até o presente momento, não existe nenhuma ratificada, pelo meio acadêmico para este fim. O estudo

elaborado, descreve quatro abordagens que atendem ao propósito do trabalho. As abordagens apresentadas são bem fundamentadas, detalhadas, recentes, e de livre acesso.

1.3. Objetivos do Trabalho

1.3.1. Objetivo Geral

O objetivo geral deste trabalho é estudar o processo de migração de dados para o *Data Warehouse*, através da descrição e análise do conteúdo de quatro abordagens com este propósito.

1.3.2. Objetivos específicos

Os objetivos específicos deste trabalho são:

a) Identificar o funcionamento da Área de estágio de dados de um *Data Warehouse*, que trata-se de uma área de organização dos dados, onde ocorre o processo de migração de dados.

b) Estudar quatro abordagens para o processo de migração de dados para o DW, verificando seus comportamentos e completudes neste processo. Tais abordagens são as seguintes:

- Uma arquitetura de extração de dados, constituída de 13 passos que engloba os processos de extração e carga de dados no *Data Warehouse*, proposta por Ralph Kimball em [KIM696].
- Um plano de conversão de dados e as especificações deste plano para abordar o processo de migração e associado a estes a preocupação na garantia da qualidade de dados, proposta por Kathy Bohn em [BOHN97].
- Estratégias de migração de dados através dos processos de perfilamento e mapeamento dos dados, proposta por John Shepherd em [SHEP699].

- A utilização de tecnologias de movimento de dados na preparação de dados para o *Data Warehouse*, proposta por Hill em [HILL98].

As quatro abordagens descritas e analisadas neste trabalho, são abordagens recentes do tema, e foram escolhidas por atenderem os propósito deste trabalho, que é um estudo sobre os conceitos envolvidos no processo de migração de dados para o *Data Warehouse*. Através do estudo das quatro abordagens, e a partir de análises, foram estabelecidas quais destas são mais completas para executar o processo. Como em todo processo de migração de dados, a qualidade dos dados deve ser garantida, foi descrito também, um processo que envolve a garantia da qualidade de dados; além de serem citadas algumas ferramentas disponíveis no mercado com estas finalidades.

1.4. Metodologia

As principais atividades deste trabalho consistem em: fazer uma coletânea bibliográfica acerca da tecnologia de *Data Warehouse*, características e estratégias utilizadas por esta. Enfatizando principalmente uma parte de sua estrutura – o *Back-End*, onde são estudados os processos de extração, transformação e carga, que compõe um processo maior denominado migração de dados para o *Data Warehouse*.

Para realizar este trabalho é necessário conhecer as diversas abordagens e conceitos sobre o processo de migração de dados para *Data Warehouse*. Para tal, será apresentado um levantamento de abordagens recentes feitas por pesquisadores ao processo de migração de dados e qualidade de dados do *Data Warehouse*. Além de serem citadas ferramentas existentes no mercado que procuram atender estas atividades. Em cada abordagem serão estudados os processos que constituem a migração de dados para o *Data Warehouse*, descrevendo as técnicas e métodos utilizados.

A análise destas tarefas compreende três fases distintas:

1) A primeira fase consiste em pesquisa bibliográfica objetivando situar e delimitar o estudo a ser analisado em um dos componentes do *Data Warehouse* – o *Back_End*, que é a parte da estrutura aonde ocorre o processo de migração dos dados*;*

2) A segunda fase é a descrição de quatro abordagens de migração de dados e qualidade de dados, identificando os processos envolvidos, os métodos utilizados, ferramentas no mercado disponíveis ;
3) A terceira fase compreende uma análise comparativa destas abordagens, verificando quais destas se apresentam mais completa em relação ao processo de migração de dados.

A primeira fase das atividades consiste em uma abordagem geral a tecnologia do *Data Warehouse*, situando em sua estrutura a área de estágio de dados, onde normalmente ocorre o processo de migração de dados. Esta fase consiste em dar base teórica e compreensão nos conceitos básicos relacionados ao *Data Warehouse* que serão utilizados nas outras fases.

A segunda fase das atividades tem como base abordagens de migração de dados e de qualidade de dados. Nesta fase são apresentados os métodos utilizados por cada abordagem para a migração de dados dos sistemas fontes para o repositório de dados, envolvendo todos os aspectos e características inerentes a cada uma. São citadas também ferramentas necessárias ao processo de migração de dados e qualidade de dados.

Com base nas informações obtidas na segunda fase, são estabelecidas análises e comparações sobre as abordagens, determinando sobre quais aspectos são diferentes e qual destas pode ser executada mais eficientemente.

1.4.1. Organização do Trabalho

O restante da dissertação é organizado de acordo com a ordem e importância em que estão apresentados os seguintes capítulos:

- O capítulo II – *Data Warehouse*, apresenta uma visão geral dos conceitos relacionados ao *Data Warehouse*, destacando arquiteturas e conceitos necessários ao bom entendimento dos capítulos subseqüentes. Começando com a evolução dos SAD’s, e descrição dos aspectos principais relacionados ao *Data Warehouse*: conceitos, arquiteturas, topologias, metodologias.

- O capítulo III – *Migração de Dados*, apresenta o processo de migração de dados para o *Data Warehouse*, contendo abordagens feitas para este processo. Descreve os aspectos principais para se ter qualidade de dados. Citando ferramentas disponíveis no mercado que visam atender o processo de migração para o DW e a qualidade de dados.

- O Capítulo IV – *Análise das abordagens de migração de dados*, apresenta uma análise entre as abordagens descritas no capítulo anterior.

- O capítulo V – *Conclusões e recomendações*, apresenta as conclusões e recomendações para trabalhos futuros.

Este trabalho apresenta dois anexos: ANEXO1 e ANEXO2, com o propósito de enriquecimento do conteúdo do trabalho, no que tange a arquitetura funcional de um *Data Warehouse* – contida no capítulo 2. Embora a arquitetura funcional contida no presente trabalho contemple as necessidades deste, julgou-se importante apresentar duas outras que estão também se destacando na construção de *Data Warehouse*.

CAPÍTULO II

Data Warehouse (DW)

2.1. Introdução

Neste capítulo, busca-se uma rápida fundamentação para o entendimento da tecnologia *Data Warehouse*, com a finalidade de compreender os capítulos subseqüentes do trabalho. O estudo inicia-se apresentando uma breve evolução dos SAD's - Sistemas de Apoio à Decisão, de onde se originou o DW; em seguida, são apresentados os conceitos relacionados à tecnologia DW, tais como, *ODS – Operational Data Store*, Modelagem multidimensional, etc. Bem como, no ambiente do *Data Warehouse* propriamente dito, onde são detalhadas suas características, componentes, dentre os quais, o *Back-End*. Onde serão descritos os tipos de atividades desenvolvidas por esta arquitetura, enfatizando seus componentes, serviços e gerenciamentos de recursos que podem ser feitos através dela. Dentre os componentes do Back_End, terá destaque a área de organização de dados, que é o lugar onde ocorre o processo de migração de dados que é o principal objeto de estudo deste trabalho.

2.2. Evolução dos SAD's

Na década de 70, surgia a primeira geração de *DSS* (*Decision Support Systems* ou SAD) que foram denominados como *MIS* (*Management Information Systems*, ou Sistemas de Informações Gerenciais), os quais eram sistemas tipo *batch* (lote) e geravam relatórios para direcionar decisões gerenciais.

Com a evolução natural dos SAD's, na década de 80, já surgiam sistemas com alguma interação com o usuário; onde se permitia fazer consultas previamente selecionadas sobre os dados. Tais consultas eram cuidadosamente realizadas, objetivando não complicar a performance dos sistemas operacionais.

O grande passo foi dado na década de 90, com o surgimento dos *EIS* (*Executive Information Systems* – Sistemas de Informações Executivas), que se baseiam em um ambiente de processamento analítico interativo e flexível, o *Data Warehousing*.

Dentre as características técnicas que qualificam um sistema como SAD tem-se [FURL94]:

a) Os SAD's tendem a ser voltados para problemas menos estruturados e menos especificados com os quais os gerentes de alto nível normalmente se deparam;
b) Normalmente, combinam o uso de modelos e técnicas analíticas com as funções tradicionais de acesso e recuperação de informações existentes nas bases de dados;
c) Buscam incluir ferramental de recursos que facilitam o uso por pessoal não qualificado em informática;
d) Enfatizam a flexibilidade e adaptabilidade para acomodar mudanças no ambiente de negócios e na abordagem de tomada de decisão utilizada pelo usuário; e
e) Devem servir de apoio a todas as etapas do processo decisório (captação do problema, elaboração de soluções e implementação).

O SAD é uma classe de sistema de informação que engloba sistemas de processamento de transações, acesso a diversas bases de dados e a uma base de modelos decisórios, e que interage com os outros componentes dos sistemas de informação no sentido de apoiar as atividades decisórias de pessoas de vários níveis.[FURL94].

2.3. Sistemas de Informações Executivas

Os Sistemas de Informações Executivas – *EIS* (*Executive Information Systems*) são sistemas que os analistas executivos utilizam para localizar problemas da corporação com precisão e que detectam tendências que são de vital importância para gerência. Tem como objetivo a análise dos aspectos que são relevantes para o funcionamento do negócio de uma empresa.

Alguns dos usos do *EIS* são [INM97]:

- Análise e investigação de tendências;
- Mensuração e rastreamento de indicadores de fatores críticos;
- Análise prospectiva;
- Monitoramento de problemas;
- Análise da concorrência.

O atendimento destes aspectos depende da existência dos dados e da atuação dos analistas como verdadeiros engenheiros de dados, que possuem preocupações tais como: procurar pela fonte definitiva de dados e dados não integrados. Mas, se existir um DW totalmente povoado e disponível, o trabalho dos analistas de *EIS* pode ser reduzido a somente análises.

O processamento *EIS* é incomparavelmente mais fácil e seguro quando operado sobre o *Data Warehouse*. Isto porque, o *Data Warehouse* possui todo alicerce para suprir as necessidades de análise em um processamento *EIS*.

A função do *EIS* [INM97]:

- Usa o *Data Warehouse* como suprimento prontamente disponível de dados resumidos;
- Usa a estrutura do *Data Warehouse* para oferecer suporte ao processo prospectivo;
- Usa os metadados do *Data Warehouse* para que analista de SAD possa planejar o modo como o sistema EIS será construído.
- Usa o conteúdo histórico do *Data Warehouse* para oferecer suporte à análise de tendências que a gerência deseja examinar.
- Usa os dados integrados encontrados ao longo do *Data Warehouse* para examinar os dados de toda a corporação.

Sistemas de Informações Executivas (EIS) versus Sistemas de Apoio à Decisão (SAD)

Os conceitos de *EIS* e SAD são muitas vezes confundidos. Apesar de estarem relacionados, o *EIS* e o SAD tratam de problemas diferenciados e, tipicamente, atendem a públicos-alvo diferentes. Um *EIS* é projetado especificamente para o uso pelos executivos, podendo estes, consultar e imprimir sem permissão de manipular os

dados. Além de ser permitido a visualização de exceções por meio de vários níveis de detalhe (*drill-down*). Já um SAD é projetado tipicamente para o nível intermediário de gerência. Os componentes básicos deste tipo de sistema incluem dados e modelos que descrevem o relacionamento dos dados; por exemplo, em um sistema de uma empresa, o campo rendimento ser obtido a partir de operações dos campos: receitas – despesas.[FURL94].

2.4. Sistemas operacionais versus Sistemas Analíticos

Existem dois tipos fundamentalmente diferentes de sistemas de informação para as organizações: sistemas operacionais e sistemas informacionais ou analíticos.

Os Sistemas Operacionais são os sistemas responsáveis pela execução de operações do empreendimento diariamente. São os sistemas da coluna vertebral de qualquer empreendimento, é método de entrada dos dados, são sistemas tais como: folha de pagamento, sistemas de contabilidade, controle de estoque, etc. Devido sua grande importância para a organização os sistemas operacionais quase sempre são as primeiras partes do empreendimento a ser computadorizada. Durante anos, estes sistemas operacionais foram estendidos e reescritos, aumentados e mantidos no ponto de serem completamente integrados na organização. De tal forma que , as maiores organizações mundiais hoje não podem operar sem os seus sistemas operacionais e os dados que estes sistemas mantêm. Mas, existem outras funções que são necessárias ao empreendimento de organizações, são funções tais como: planejamento, previsão e administração da organização. Estas funções também são críticas à sobrevivência da organização, especialmente em nosso mundo atual de ritmo veloz [ORR97]. Funções como, por exemplo, "planejamento de marketing", "planejamento de criação" e "análise financeira" necessitam de suporte de sistemas de informação, e por tratarem-se de funções diferentes da operacional, requerem sistemas e informações também diferentes, para tais funções, os sistemas informacionais ou analíticos são os utilizados por darem suporte as funções baseadas em conhecimento.

Os Sistemas Analíticos têm haver com analisar dados e tomar decisões, normalmente a respeito de grandes decisões sobre como o empreendimento vai operacionalizar no presente e no futuro. Estes sistemas não têm somente um enfoque

diferente do operacional, eles têm freqüentemente um campo de ação diferente. Enquanto os dados operacionais necessitam ser normalmente enfocados em uma única área, dados analíticos necessitam freqüentemente ser estendidos a várias áreas diferentes e necessitam de grandes quantidades de dados operacionais relacionados.

Estes tipos de sistemas possuem a característica de não serem voláteis, eles não atualizam continuamente as informações, e as mantêm como snapshot (instantâneo) de dados, que é um registro específico do tempo, ocasionando um grande armazenamento de dados históricos. E por esta característica são denominados sistemas somente de leitura, ao passo que os sistemas operacionais são denominados de escrita e leitura, por executarem as atividades básicas de inserção, atualização, consulta e deleção de dados em um banco de dados operacional.

2.5. Modelagem Dimensional

A modelagem dimensional é uma alternativa para a MER - Modelagem Entidade e Relacionamento, além de possuir as mesmas informações que esta contém [KIM98L].

A MER é uma técnica de projeto conceitual que através da criação de entidades e relacionamentos diferentes, procura eliminar a redundância de dados [KOR95].

Apesar da MER ser excelente para construção de sistemas OLTP pelas características que apresenta, o mesmo não ocorre em relação aos sistemas OLAP, por causa das restrições que esta modelagem apresenta, que são desfavoráveis a estes sistemas. A modelagem dimensional baseia-se na representação de quase todos os tipos de dados do negócio por um tipo de cubo de dados, no qual possui células com valores medidos e os lados do cubo definem as dimensões dos dados. Este cubo pode ter mais que três dimensões é tecnicamente chamado de hipercubo, apesar de geralmente os termos cubo e cubo de dados serem usados como sinônimos de hipercubo [KIM98L].

O modelo dimensional apresenta a capacidade de todas as tabelas existentes poderem ser modificadas localmente pela simples adição de novas linhas de dados na tabela, e as ferramentas de consulta e geradoras de relatórios não necessitarem de reprogramação para acomodar as modificações; e todas as aplicações continuarem a executar sem proporcionar diferentes resultados. Desta forma, é possível acomodar novos e inesperados elementos de dados e novas decisões de projeto.

Em bancos de dados analíticos que manipulam multidimensões, existem dois tipos principais de esquemas que normalmente são utilizados: o esquema estrela *(star schem*a) e o esquema floco de neve *(snowflake schem*a). E para quaisquer destes esquemas são utilizados basicamente dois tipos de tabelas:

- Tabelas fato - um fato usualmente é algo sobre o qual não se conhece antecipadamente; é uma observação da realidade [KIM98T]. As tabelas fatos normalmente armazenam grande quantidade de dados e são centrais. Dependem diretamente da granularidade adotada; possuem chaves primárias compostas e contêm as medições numéricas do negócio, denominados fatos. Os melhores fatos e os mais úteis são numéricos e possuem como características o fato de serem diferentes a cada medida e de modificarem-se a cada combinação de atributos das tabelas dimensão.

- Tabelas dimensão: os componentes da tabela descrevem as características de uma coisa tangível. As tabelas dimensão são simétricas em relação à tabela fato, geralmente possuem uma chave primária simples e campos denominados atributos; armazenam pequena quantidade de dados, quando comparadas com a tabela fato e contêm os dados descritivos do negócio. A cada chave primária da tabela dimensão corresponderá exatamente a uma chave estrangeira na tabela fato, permitindo assim a ligação entre ambas. Apesar de muitas vezes conterem campos numéricos, a diferença em relação aos fatos é que nas dimensões estes são constantes, ou melhor, não variam continuamente a cada nova amostra [KIM98T].

2.6. *Data Mart* (DM)

Data Mart é um conceito importantíssimo dentro da tecnologia de *Data Warehouse*. Os vários autores que escrevem sobre este assunto normalmente apresentam uma grande quantidade e diversidade de definições, tais como:

Data Mart é um subconjunto lógico de um completo *Data Warehouse*. Um *Data Warehouse* é composto da união de todos seus *Data Mart* . O *Data Mart* é normalmente organizado ao redor de um único processo de negócios.[KIM98T].

Os *Data Marts* são *Data Warehouses* menores, usualmente enfocando em um subconjunto pequeno dos dados de empreendimento [SRIV99]. Tipicamente cada *Data Mart* é usado por uma unidade particular da organização para várias análises estratégicas pertinente as suas metas. Dados são extraídos do *warehouse* incorporado nos *Data Marts* periodicamente, e usados para análise. Resposta interativa é um assunto em *Data Mart*, quando ferramentas de análise interativas trabalham diretamente nos dados.[SRIV99].

Os Data Marts tratam de problemas empresariais específicos. Ao contrário de *Data Warehouse*, o negócio de caso de um *Data Mart* é sólido antes de começar o desenvolvimento, a comunidade de usuário é anteriormente conhecida e normalmente concentrada em um ou dois departamentos, e os gerentes de negócio têm uma aplicação e freqüentemente uma ferramenta de *front-end* específica já em mente[DYCH98].

Um *Data Mart* é uma coleção de bancos de dados e ferramentas enfocadas em um problema empresarial específico. Tamanho somente não define *Data Mart*, entretanto eles tendem a ser menores que *Data Warehouse* que são grandes coleções de dados de empreendimento que compreendem numerosas áreas e tópicos empresariais [MALL99].

Os altos custos de implementação de um DW limitam o seu uso por grandes companhias, as quais muitas vezes não estão dispostas a correr riscos no investimento em um empreendimento que não se tem certeza do sucesso e, consequentemente, o retorno do investimento, tornando os DM, nesse caso, uma alternativa reduzida e de baixo custo[GRA98].

A construção de um *Data Mart* é muito subjetiva, pelo fato dele ser construído pelas empresas de acordo com o conjunto de exigências delas.

Abordagens do Data Mart

As abordagens utilizadas na construção de DM são as abordagens *top-down* e o *bottom-up*. A abordagem *Top-down* consiste em iniciar a partir de um *Data Warehouse* e gerar *Data Marts* deste repositório central. Já a abordagem *bottom-up* seria o contrário da anterior, isto é, *Data Marts* seriam construídos em direção a algum tipo de *Data Warehouse*.

Normalmente, os analistas recomendam a abordagem *top-down*, por ser uma abordagem que dá flexibilidade e a habilidade para popular novamente os dados quando

mudanças acontecem. Nesta abordagem é possível apagar um *Data Mart* e substituí-lo por um novo, pelo fato dos dados que estão povoando o *Data Warehouse* já terem sido limpos, integrados e transformados.

Embora para maioria dos analistas a abordagem *top-down* ser a ideal, nem sempre acontece deste modo. Companhias podem querer a abordagem *bottom-up* se simplesmente necessitam de um *Data Mart* para resolver um problema específico e não querem se antecipar em construir um *Data Warehouse* repleto de vazios.

Independente da abordagem escolhida, o importante é assegurar que os *Data Mart* se comuniquem. Pois *Data Mart "stovepipe*" são inaceitáveis em um projeto. Logo, para todo *Data Mart* construído independente de um DW, deve ter sempre assegurada a sua consistência.

Data Marts empacotados

As empresas decidindo montar um *Data Mart*, podem comprar um *Data Mart* empacotado ou podem construir um. Estes pacotes geralmente são acompanhados de software de transformação, ferramentas de administração de metadados e algumas ferramentas de consultas.

Os pacotes de *Data Mart* podem ser genéricos ou de aplicação específica. Como exemplo de pacotes de *Data Marts* genéricos especialistas citam: *Data Mart Suite* da *Oracle Corp's* e *Visual Warehouse* da *IBM*. Eles oferecem alguma movimentação de dados, ferramentas de acesso e ferramentas de banco de dados e consulta.

É possível também comprar um pacote de solução para endereçar uma aplicação particular. A aquisição de pacotes de *Data Mart*, merece especial atenção, isto porque estes pacotes assumem que a empresa não objetará caso o banco de dados seja diferente do banco de dados padrão. Além de que eles assumem que todos os dados da empresa agem juntos, e que estão em boas condições e que a empresa não possui muitos dados codificados manualmente.

É importante saber que a aquisição de pacotes de *Data Mart* deve ser bastante significativa para a organização. Pois quando estes são inteiramente adaptáveis aos dados destas, ocorre uma grande economia de despesas que estas teriam com recursos e tempo disponível ao projeto, caso fossem construir o *Data Mart* completo.

Comparações entre Data Warehouses e Data Marts

DATA WAREHOUSES VS. DATA MARTS		
Campo de ação	• Aplicações neutras • Centralizado, distribuído • Compatível com várias linhas de negócio corporativo • Arquitetura	• Requisitos de aplicações específicas • LOB, departamento ou área de uso • Negócio orientado a processo • Vários bancos de dados com dados redundantes
Perspectiva de dados	• Dado histórico detalhado • Alguns sumarizados • Levemente desnormalizado	• Detalhado (alguns históricos) • Sumarizados • Altamente desnormalizados
Domínio	• Áreas múltiplas de assunto	• Área de assunto parcial única • Área de assunto parcial múltiplas • instantâneos de fonte operacional
Fonte de Dados	• Muitas • Dados externos, operacionais	• Poucas • Dados externos, Operacionais. • instantâneos de banco de dados OLTP • Extração de dados fabricada
Tempo de implementação de estrutura	• 9 à 18 meses para primeira fase(duas ou três áreas de assunto) • Implementação de armazenagem múltipla	• 4 a12 meses
Características	• Flexível, extensível • Orientado a dados • Durável/estratégico	• Vida curta/tática • Restrito, não extensível • Orientado a projeto

TABELA 1 - *DATA WAREHOUSE* VERSUS *DATA MARTS*.

O *Data Mart* é a restrição do *Data Warehouse* para um único processo de negócios ou para um grupo de processos de negócios designado para um grupo particular de negócios [KIM98T].

Embora estejam intrísecamente ligados, *Data Mart* e *Data Warehouse* possuem características próprias, como mostradas na tabela 1, que apresenta as características de ambos para estabelecer comparações de aplicabilidade do uso destes. Na coluna 1, temos os aspectos sobre os quais o *Data Warehouse* e o *Data Mart* são analisados. Na coluna 2, temos os aspectos do *Data Warehouse* e na coluna 3, temos os do *Data Mart*.

O estabelecimento do uso de *Data Warehouse* e *Data Mart*, depende do que a corporação que vai utilizá-lo pretende fazer, levando-se em consideração os fatores de investimentos da empresa neste setor, recursos financeiros disponíveis , custos e o tempo e recursos humanos que o projeto vai envolver para ser concluído.

2.7. *ODS* – Operational Data Store

O Armazenamento de Dados Operacionais – *ODS* (*Operational Data Store*), constitui a base do processamento informacional operacional, e seu uso destina-se aos gerentes que precisam tomar decisões instantâneas.

No ODS os dados são atuais ou quase atuais e são dados passíveis de atualização. Por isso , pode prover uma integrada, visão coletiva operacional de informação.

O ODS possui características, tais como: integração de alto desempenho e dados muito atuais e detalhados.

O ODS pode estar presente ou não em uma arquitetura. Por exemplo uma arquitetura pode ser composta por um ambiente legado, um *Data Warehouse* e um ODS, ou somente pelos dois primeiros itens, e estes dois tipos de arquiteturas estarão corretas. O motivo é a formalidade que alguns sistemas dão a arquitetura sem ODS ou com ODS. É importante frisar que a existência de uma arquitetura não invalida a existência da outra.

O ODS possui uma arquitetura semelhante ao *Data Warehouse*, pois ambos são integrados e orientados por assunto. Mas diferenciam-se muito no aspecto de atualização e uso. O *Data Warehouse* é não volátil, não sofre atualizações através de

transações, e é utilizado para decisões estratégicas, enquanto que o ODS pode ser atualizado através de transações e seu uso é para decisões táticas.

Existem tipos diferentes de ODS [INM99]:

ODS classe 0 – é um tipo de ODS onde são copiadas tabelas inteiras em uma base de dados específica do ambiente operacional. Apesar de ser fácil construir, é uma forma muito fraca de ODS, não há nenhuma integração de dados. E tende a ter um ciclo de vida muito pequeno.

.

ODS classe I – é um tipo de ODS de fácil construção, onde as transações são transportadas para ele de uma maneira rápida e por atacado. Neste caso, a transação executa no ambiente operacional, e logo em seguida passa para o ODS. Dados não são integrados e não são ajustados para serem colocados no *Data Warehouse*.

ODS classe II ou III - é um ODS onde os dados realmente atravessam uma camada de integração e transformação. Neste caso, os dados são verdadeiramente integrados. O que decorre um atraso para que os dados sejam refletidos no ODS. Tem difícil construção em decorrência de que programas de integração e construção também devem ser construídos. Porém, os dados ficam ajustados para entrar no *Data Warehouse* após passar pela integração e transformação.

ODS classe IV – é um tipo de ODS alimentando por dados analíticos agregados do *Data Warehouse*. O analista de DSS examina um corpo de dados no *Data Warehouse* e decide que dados poderiam ser úteis para a companhia interagir com sua base de cliente. Então estes dados são passados ao ODS. Desta forma, uma corporação tem acesso a dados analíticos *online* em um modo de tempo real. Esta interface é fácil construir desde que o *Data Warehouse* já tenha sido construído.

ODS resultante da combinação de classes – é um tipo de ODS onde há uma combinação de dados integrados do ambiente operacional e dados agregados do ambiente analítico. Este é o ODS mais poderoso e é o mais difícil de construir. Ele é uma combinação da Classe II, III e IV de ODS. É a forma mais poderosa de ODS.

Todos estes conceitos anteriores tem ligação intrínseca e necessárias ao bom entendimento do *Data Warehouse*. Nas próximas seções serão abordadas as metodologias, arquiteturas e definições referentes a ele.

2.8. Data Warehousing

Data Warehousing, refere-se a uma coleção de tecnologias que objetivam melhorar a tomada de decisões, desta forma, trata-se do processo e não do produto. Este processo gerencia as informações para tomada de decisões das empresas através de: modelagem conceitual dos fatos do negócio, modelagem física, processo de extração de dados de sistemas existentes, limpeza, transformação e carga em um repositório de dados, o *Data Warehouse* (DW) – que tem como público alvo os tomadores de decisões gerenciais de alto nível e a longo prazo. O DW apresenta duas características essenciais: ser integrado e histórico, proporcionando respectivamente bases para análise de tendências e observação do andamento da empresa por um grande espaço de tempo.

Os elementos de dados brutos que são transformados para povoar o DW, são oriundos de dados operacionais e das fontes externas (por exemplo, os sistemas "legados" – ambiente operacional não integrado). A extração destes dados é feita através de ferramentas específicas, onde se destacam, as ferramentas *OLAP* (*Online Analytic processing*), que proporciona ao usuário grande capacidade de manipulação dos dados e análise crítica dos resultados obtidos.

É de consenso entre os pesquisadores do assunto, como [KIM98L] que o ambiente de um DW envolve três grandes componentes:

- *Back End*
- O repositório *Data Warehouse*
- *Front End*

O *Back End* compõe a arquitetura de dados onde o processo de transformação de dados ocorre. Tem como principal objetivo agregar e conduzir os dados corretos de um

lugar a outro, utilizando os subprocessos de extração, limpeza, sumarização e carga dos dados.

O *Data Warehouse* é o alicerce do processamento analítico de verificar dados [INM97]. No ambiente analítico, os dados destinados para a análise do usuário final, são obtidos a partir de uma fonte única (DW ou *Data Mart* (DM)), tornando a tarefa do analista de SAD mais fácil e precisa que no ambiente operacional, onde as diversas fontes são independentes e os dados não estão integrados. O DW é um banco de dados analítico projetado especificamente para suporte à decisão, onde os dados desnormalizados permitem melhor desempenho na recuperação de informações.

O *Front End* é a face pública do *Warehouse*. É o que é visto e no qual o usuário final trabalha normalmente realizando consultas – é a interface do usuário.

Nos últimos anos, *Data Warehousing* se desenvolveu rapidamente de um conjunto de idéias relacionadas em uma arquitetura para transmitir dados para usuário final de empreendimento computacional.

2.9. Ambiente do *Data Warehouse*

2.9.1. Definição de *Data Warehouse*

Data Warehouse é uma coleção de dados orientados por assuntos, integrado, não volátil e variável em relação ao tempo, objetivando dar suporte aos processos de decisão [INMON92].

Nesta definição têm-se embutidas as principais características do DW, que são especificadas da seguinte forma:

2.9.2. Características

Orientado por assunto – O DW organiza seus dados pelos assuntos inerentes à organização o qual se destina, desta forma, os dados são organizados pelos temas que são de interesse aos analistas de SAD da organização.

Integrado – O DW é integrado. Os dados devem ser aceitos de uma forma global, logo deve ter padrão único, mesmo quando aplicações operacionais dos quais são oriundos, os armazene de maneira diferente.

Não-volátil – No DW ocorrem somente dois tipos de operações sobre os dados: a carga inicial e o acesso aos mesmos, logo não existem atualizações sobre os dados armazenados.

Variante em relação ao tempo – Os dados do DW em algum instante do tempo tem precisão, logo, diz-se que estes dados têm variação em relação ao tempo. Esta variação pode apresentar-se de várias maneiras:

- Os dados representam informações sobre um horizonte de tempo muito amplo de cinco a dez anos.
- Cada estrutura básica no DW contém implícita ou explicitamente, um elemento de tempo como dia, semana e mês.
- Os dados do DW, quando armazenados corretamente, não podem ser atualizados. Estes dados são disponibilizados somente para leitura.

2.9.3. Granularidade

A granularidade corresponde ao nível de detalhamento ou resumo dos dados existentes no DW. Quando os dados estão muito detalhados, significa que existe um baixo nível de granularidade. Ao passo que, quando estes são pouco detalhados, existe um nível alto de granularidade. Por análise, pode-se verificar que o volume de dados do DW depende diretamente do nível de granularidade que lhe é destinada.

Além do volume de dados, outro importante aspecto que sofre interferência da granularidade é a questão da consulta aos dados que o DW pode atender. Em DW com nível alto de granularidade, as consultas que são atendidas têm um aspecto geral, logo consultas que requerem grandes detalhamentos, não serão contempladas adequadamente para DW com esta característica.

O nível de detalhe de uma consulta está intrinsecamente relacionado ao volume de dados. Para consultas muito detalhadas, tem-se um baixo nível de granularidade, logo o

espaço para armazenar dados é grande - DW com grande volume de dados. No caso de consultas mais resumidas, tem-se um alto nível de granularidade, que requer menos espaço para armazenar dados, o que implica em menos bytes e menos índices.

Alguns aspectos devem se analisados quando à questão da granularidade em um DW ser alta ou baixa. Um DW com alto nível de granularidade, possui um atendimento limitado as consultas, porque como os dados encontram-se resumidos não conseguem fornecer informações mais detalhadas. Já um DW com baixo nível de granularidade, pode atender satisfatoriamente todas as consultas possíveis sobre os dados, embora ocasionando um grande volume de dados e processamento não eficiente. Em se tratando de SAD a melhor alternativa é o DW com alto nível de granularidade, pois como bem escreveu [INMON97]: durante o processamento de SAD, como é comum no ambiente de *Data Warehouse*, dificilmente um evento isolado é examinado. É mais comum ocorrer à utilização de uma visão de conjunto dos dados.

Para *Data Warehouse* com grande volume de dados, o nível dual de granularidade no nível detalhado é a melhor opção. O procedimento adotado é estabelecer para o DW dois tipos de dados armazenados, os dados levemente resumidos e os dados históricos detalhados. O objetivo é fazer com que o processamento executado mais vezes pelo SAD ocorra nos dados levemente reduzidos, uma vez que estes estão compactados e são de fácil acesso. Não deixando de contemplar também as consultas mais detalhadas, uma vez que existirá também o nível de dados históricos detalhados.

Em se tratando de nível detalhado de dados esta é a melhor opção arquitetônica para o DW, por proporcionar possibilidade de atender a qualquer nível de consulta que possa ser respondida, com eficiência, facilidade de acesso e custos razoáveis. A aplicabilidade de um único nível de detalhe de dados só ocorreria para DW's que possuem volume de dados relativamente baixo.

2.9.4. Agregação

A agregação corresponde a sumarização de que estão logicamente redundantes com os dados que já povoam o *Data Warehouse*, e são utilizados para aumentar enormemente o desempenho das consultas.[KIM98L].

Segundo [INM97]: um registro de agregação é criado pelo agrupamento de diversos registros detalhados. Em um projeto de banco de dados analítico, a agregação de dados operacionais em um único registro pode ter várias formas, tais como:

- Os valores provenientes dos dados operacionais podem ser resumidos, totalizados ou processados para se obter o ponto mais alto, mais baixo, média, etc.

- Dados de tipos predeterminados, que estejam dentro de limites, podem ser medidos.

- Dados válidos em relação a um determinado momento podem ser dispostos em um bloco.

Quase todos os *Data Warehouses* contém agregados pré-calculados [KIM98L], que possuem objetivos básicos tais como: melhorar o tempo de respostas de consultas para o usuário final. E reduzir o espaço de armazenamento, uma vez que agrupamentos de vários dados estarão em um único registro. [INM97].

O processo de agregação no *Data Warehouse* possui a desvantagem de possibilitar a redução da capacidade ou funcionalidade do *Data Warehouse*, ocasionando a perda de detalhes. O grande trabalho do projetista é garantir que os detalhes perdidos não sejam de extrema importância. Para tal são utilizados dois processos: o primeiro consiste em criar dados agregados iterativamente com o usuário e o segundo método consiste em garantir que nenhum detalhe importante seja perdido durante a construção do processo de agregação [INM97].

2.9.5. Estrutura do *Data Warehouse*

A figura 2, apresenta a estrutura interna do DW, que possui os seguintes componentes sobre os dados:

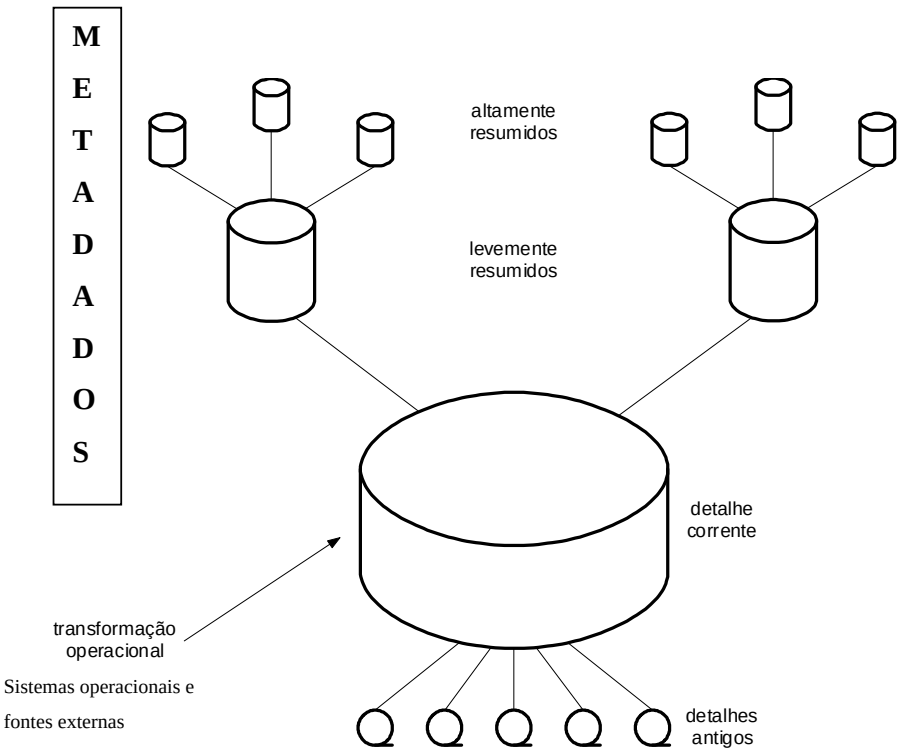

FIGURA 2 – A ESTRUTURA INTERNA DO *DATA WAREHOUSE*

Nível de detalhe dos dados antigos – tem nível de detalhe consistente com os dados detalhados atuais, são pouco acessados. São armazenados geralmente em fitas magnéticas.

Nível de detalhe dos dados correntes – refletem acontecimentos recentes, armazenados em discos e no nível mais baixo de granularidade.

Nível de detalhe dos dados levemente resumidos (o departamental ou Data Mart) – são resultantes da sumarização do detalhe de baixo nível encontrado nos dados detalhados atuais.

Nível de detalhe dos dados altamente resumidos (o individual) – são compactados e de acesso fácil.

Na passagem do ambiente operacional para o *Data Warehouse* os dados sofrem significativas transformações.

Os dados operacionais depois de transformados são armazenados no nível de detalhe corrente. E quando os dados são considerados antigos, passam do nível de detalhe corrente para o nível de detalhe antigo. À medida que os dados são resumidos, passam do nível de detalhe corrente para o nível de detalhe levemente resumido e, do nível de detalhe levemente resumido para o nível de detalhe altamente resumido.

2.9.5.1. Metadados

Os metadados são dados acerca dos dados, sendo utilizados como:

i. Diretório auxiliar na localização de componentes do DW.

ii. Guia para o mapeamento dos dados em sua conversão desde o ambiente operacional até o ambiente do DW.

iii. Guia para algoritmos utilizados para sintetização entre o níveis dos dados.

Os metadados podem ser classificados em [BER97]:

Metadado técnico – contém informação sobre o armazenamento dos dados, tais como: informação sobre a origem dos dados, descrição das transformações, objetos de armazenamento e definições de estruturas de dados para dados requisitados, regras usadas para realizar a limpeza dos dados e o crescimento dos dados,operações de mapeamento dos dados quando a captura ocorre nos sistemas originais e aplicá-las para os propósitos do banco de dados do *Data Warehouse*, autorização de acessos, históricos de backup, e tudo o que é de interesse dos administradores do *Data Warehouse*.

Metadados de negócios – Contém informações para dar uma visão fácil ao usuário sobre a informação armazenada no *Data Warehouse*. As informações são: áreas de interesse, páginas da internet, outras informações para dar suporte a todos os componentes do *Data Warehouse*, informações operacionais sobre o *Data Warehouse*.

2.9.6. Metodologia de desenvolvimento de um DW

Embora não exista no meio acadêmico, metodologia de desenvolvimento de DW consagrada. Será apresentada nesta seção a metodologia proposta por [KIM98L] por caracterizar e detalhar bem as fases de planejamento, levantamento de requisitos, arquitetura funcional, projeto da base de dados, aplicações de usuários finais, auditoria nos dados e uso, suporte e extensão do DW.

O ciclo de vida proposto por [KIM98L] está esquematizado na figura 3, onde se deve considerar a existência de uma seqüência temporal entre as fases apresentadas, por exemplo: executar 'planejamento' antes de 'definição de requisitos do negócio'. Mas algumas fases podem ser executadas concorrentemente. Por exemplo: 'projeto da arquitetura técnica' e 'modelagem dimensional'.

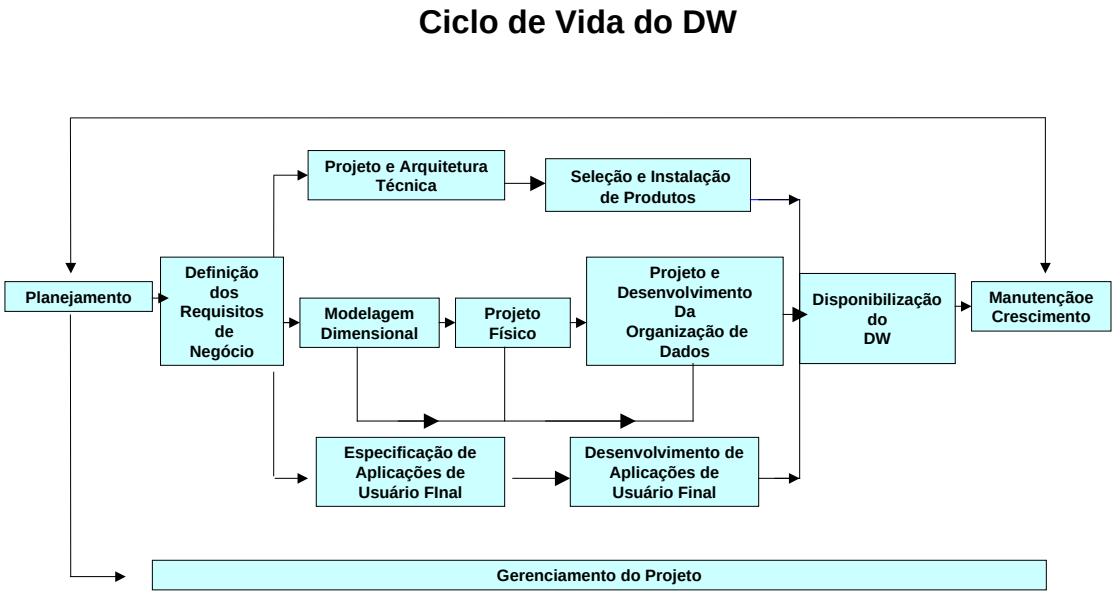

FIGURA 3 – CICLO DE VIDA DO DW

O ciclo do desenvolvimento do DW começa com a fase de planejamento. As atividades de planejamento variam bastante de uma corporação para outra. O planejamento do DW é a primeira atividade crítica do DW e uma das mais importantes, considerando que as qualidades dos levantamentos e definições afetarão o projeto como

um todo. Por esse motivo deve-se ser o mais criterioso possível nos diversos levantamentos, analisando o máximo de condicionantes do ambiente e transformando as definições em um documento denominado de Plano de Projeto ou algo similar, que servirá como "mapa" durante a execução do projeto. Esses planos por ocasião da construção do DW são cumpridos, acompanhados pelos diversos responsáveis e reavaliados na medida da execução do projeto de DW. As principais atividades que envolvem a fase de planejamento do projeto e seu gerenciamento, são as seguintes:

- Definição do projeto, através da verificação de real demanda por informação gerencial e de onde ela está (por exemplo: grupos de chefes e lideres influentes na organização desejosos pela implementação do DW, motivação para o projeto de DW, cultura organizacional, viabilidade técnica, parceria entre usuários e o departamento responsável pela informação na organização).
- Definição do escopo inicial do projeto, baseado nos requisitos do negócio, o qual é feito através da realização de entrevistas com as diversas classes de usuários.
- Levantamento dos fatores que permitem mensurar o sucesso, assim como dos riscos do projeto do DW.
- Elaboração de justificativas para o projeto, abordando-se o investimento financeiro e seus custos (por exemplo: software, hardware, manutenção, recursos externos necessários, etc.), assim como o retorno financeiro e seus benefícios.
- Elaboração de um Plano de Projeto, a partir dos dados iniciais levantados. Além de registrar os dados já descritos nesta seção, este documento deve conter outros dados, tais como: identidade (nome) do projeto; identificação do pessoal participante (por exemplo: patrocinadores, gerente de projeto, analistas de negócios, encarregados pela modelagem de dados, DBA, desenvolvedores de aplicações, arquitetos de dados, especialistas em suporte, programadores.), assim como as tarefas correspondentes, atribuições e prazos a serem cumpridos no desenvolvimento do projeto de DW; identificação dos demais recursos do projeto necessários (por exemplo: hardware, software, treinamento, assessoria.); estabelecimentos de data de início do desenvolvimento do projeto; especificação de cronograma das diversas fases e atividades do DW, assim como o status indicando a porcentagem do trabalho realizado em relação ao total; plano de comunicação entre os participantes do projeto; plano contendo os critérios de medição de sucesso do projeto de DW e identificação de

dependências entre tarefas, ou seja, fazer levantamento de tarefas que devem ser concluídas antes de se iniciar outras.

A fase *definição de requisitos do negócio* aborda a importância de se entender os fatores fundamentais que dirigem o negócio do usuário, tendo em vista que estes trarão impactos diretos em todas as demais fases e aspectos do DW. É imprescindível entender com clareza o negócio do usuário final, suas particularidades, requisitos e exigências, cujos dados podem ser levantados através de entrevistas, seções de facilitação, análise de documentos e relatórios, auditoria nos dados internos e externos da corporação ou ainda através de outros métodos quaisquer a serem conduzidos pela equipe de projeto sobre as diversas classes de usuários devidamente selecionadas.

Nesta fase são realizadas as seguintes atividades:

- Identificação e preparação da equipe de entrevistadores; seleção de entrevistadores para cada entrevista; agendamento e realização das diversas entrevistas, procurando ao final de cada uma identificar os possíveis fatores que permitem mensurar o sucesso do DM; análise dos resultados de cada entrevista em reunião realizada com a presença de todos os entrevistadores e documentação dos principais aspectos das diversas entrevistas.
- Revisão do escopo do projeto e priorização.
- Revisão do Plano de Projeto.

A fase *projeto da arquitetura técnica* é uma das mais importantes no projeto do DW. Nesta fase são realizadas duas atividades fundamentais: uma consiste no estabelecimento de uma estrutura arquitetural de alto nível, também referenciada como arquitetura funcional, onde são abordados os elementos fundamentais em um DW, tais como a área interna, área externa e assim como os serviços de ambas as áreas (Figura 7), e a outra consiste na especificação da infra-estrutura técnica e os respectivos componentes necessários para permitir a criação do DW, de acordo com o especificado na arquitetura técnica. Ressalta-se novamente que a arquiteturas técnica e infra-estrutura estão intimamente ligadas entre si, ou seja, os componentes e tecnologias da infra-estrutura dependerão diretamente do projeto de arquitetura técnica adotado.

As atividades que compreendem esta fase são as seguintes:

- Especificação de um grupo responsável pela criação da arquitetura do DW.

- Reunião e documentação dos requisitos técnicos; revisão do ambiente técnico corrente; criação de uma arquitetura técnica e determinação das etapas a serem cumpridas na arquitetura técnica.

- Criação de um plano de infra-estrutura, definindo hardware, software, servidores, rede de comunicações, estações de trabalho, etc.

- Aceitação do projeto pelo usuário final.

- Revisão do projeto.

A fase de *seleção e instalação de produtos* somente deverá ser executada após a criação do projeto de arquitetura técnica e consiste na realização das seguintes atividades:

- Pesquisa de produtos candidatos.

- Avaliação das funcionalidades dos produtos sob diversos pontos de vistas, como por exemplo, funcionalidades básicas, tais como: possibilitar a extração de múltiplas plataformas e fontes de dados, prover compressão e descompressão de dados, suportar funções de transformação; controle de trabalho e agendamento (por exemplo: suporta agendamento baseado em tempo e/ou evento, realiza monitoramento, suporta arquivamento e restauração de dados, provê sincronização em atualizações), metadados e padrões (por exemplo: disponibiliza repositório aberto, suporta o emprego de metadados com produtos de terceiros, suporta diversos níveis de transporte de dados - TCP/IP - FTP), itens específicos do vendedor (por exemplo: custo, suporte técnico, documentação, treinamento, consultoria).

- Desenvolvimento de protótipos pelos vendedores, em um ambiente devidamente controlado e parametrizado (por exemplo: mesmo problema, requisitos, dados, tempo), de modo a permitir uma melhor avaliação das funcionalidades dos softwares pré-selecionados para aquisição. É conveniente que as avaliações sejam conduzidas sob os aspectos previstos no planejamento da arquitetura e seus serviços (por exemplo: área interna, área externa e seus serviços).

- Seleção das ferramentas e produtos mais adequados e estabelecimento de contrato com vendedores.

- Aceitação do projeto pelo usuário final.

- Revisão do projeto.

Na fase de *modelagem dimensional,* os requisitos do negócio e os dados necessários para atender as exigências analíticas, levantadas por ocasião da fase "definição de requisitos do negócio", são utilizados para desenvolver um modelo de dados dimensional adequado.

Nesta fase são realizadas as seguintes atividades:

- Construção de uma matriz onde um dos lados representa todos os DM passíveis de desenvolvimento, e o outro lado todas as possíveis dimensões. A interseção entre os dois lados, possibilita a visualização das dimensões de determinada área de negócio (DM). Um dos objetivos dessa matriz é assegurar que o DW seja utilizável e extensível a todas as áreas da organização.
- Definição do DM a ser desenvolvido e as correspondentes granularidade de dados, tabelas dimensão e seus atributos, tabelas fato e seus fatos, agregados.
- Revisão da modelagem dimensional com o usuário final e aceitação do modelo de dados.
- Revisão do projeto e recomendações quanto às ferramentas de usuário final e à base de dados analítica..
- Atualização do projeto lógico da base de dados analítica.
- Revisão lógica do projeto da base de dados analítica.
- Aceitação do projeto pelo usuário final.
- Revisão do projeto
- Durante a fase de modelagem dimensional também é realizada a análise das diversas fontes de dados de modo a identificar aquelas que possuem os dados necessários para atender o modelo de dados, assim como procurar especificar dentre estas as melhores fontes de dados. Essa análise compreende as seguintes atividades:
- Identificação das fontes de dados candidatas, verificando detalhes tais como sistema *OLTP* que integra, plataforma (por exemplo: *UNIX, Windows NT*, etc.), estrutura física dos arquivos (por exemplo: *flat files, Oracle, Excel*), volume de dados (por exemplo: número de transações por dia, média de transações por semana, etc.), identificação de chaves primárias e estrangeiras e características dos campos de dados (por exemplo: tipos de dados, comprimento, precisão).
- Estudo do mapeamento de dados das fontes candidatas para as tabelas de destino dos dados.

- Estimativas do número de linhas ou registros das fontes de dados candidatas.
- Revisão e aceitação pelo usuário final.
- Revisão do projeto.

Na fase de *projeto físico* são realizadas as seguintes atividades:

- Definição de nomes padronizados para os objetos da base de dados.
- Execução do projeto físico, criando os objetos da base de dados analítica.
- Estimativa do tamanho da base de dados analítica. Como regra de estimativa geral, um completo DW necessita tipicamente de um espaço de armazenamento equivalente a três ou quatro vezes o espaço de dados requeridos pelos dados atômicos do esquema estrela.
- Desenvolvimento de um plano inicial de indexação, agregação e particionamento.
- Aceitação do projeto pelo usuário final.
- Revisão do projeto.

A fase *projeto e desenvolvimento da área de organização de dados*, diz respeito as atividades fundamentais a serem realizadas no DW: a extração, transformação e carga de dados.

As atividades que compreendem esta fase são as seguintes:

- Criação de uma arquitetura de alto nível que represente o fluxo de dados dos sistemas fonte para a base de dados analítica de destino.
- Teste e escolha de ferramentas de terceiros ou desenvolvimento de programas específicos para a realização das atividades de organização de dados.
- Detalhamento da arquitetura de alto nível (por exemplo: determinando por exemplo quais tabelas e em que ordem deverá ser realizada as atividades de extração, transformação e carga; quais as atividades de transformação que serão realizadas nas diversas tabelas).
- Realização do processo de organização de dados com uma dimensão e avaliação do resultado.
- Definição e realização de modificações lógicas e atualização dos registros das dimensões, quando necessário. São propostas três técnicas básicas quando um registro da dimensão é atualizado: substituição pelo registro mais recente, criação de novo registro na dimensão ou, por último, criação de um novo campo na dimensão que

armazene somente o campo alterado, de forma que sejam armazenados os campos novos e antigos.

- Realização do processo de organização de dados com as demais dimensões.
- Realização do processo de organização de dados com a tabela fato.
- Desenvolvimento de procedimentos que permitam a carga incremental de tabelas fato que sejam muito grandes, utilizando os recursos baseados em novas transações, "*logs*" de bancos de dados, replicação, realização de múltiplos passos de carga e execução paralela.
- Carga de tabelas de agregados.
- Operação e automação do processo de carga, podendo incluir atividades, tais como: agendamento de trabalhos, monitoramento, geração de "*logs*", manipulação de exceções e erros, notificação.
- Desenvolvimento e aplicação de procedimentos para assegurar a qualidade dos dados.
- Desenvolvimento e realização de atividades de arquivamento, backup e recuperação de dados.
- Aceitação do projeto pelo usuário final.
- Revisão do projeto.

A fase *especificações de aplicações de usuário final* procura identificar as áreas prioritárias e, a partir destas, definir um conjunto padronizado de aplicações destinadas aos usuários finais, uma vez que não são todos os usuários que necessitam ter acesso "ad hoc" aos dados do DW. As atividades que compreendem esta fase são as seguintes:

- Priorização e identificação dos relatórios candidatos; desenvolvimento de uma estrutura geral que permita aos usuários acessar os diversos relatórios de forma estruturada (por exemplo: menu de relatórios) e determinação de estruturas padronizadas de relatórios para os usuários finais.
- Revisão das estruturas padronizadas e documentação.
- Aceitação do projeto pelo usuário final.
- Revisão do projeto.

Na fase de *desenvolvimento de aplicações para usuário final* são desenvolvidas as aplicações necessárias de acordo com levantamentos realizados na fase de "especificações de aplicações de usuário final". As atividades que compreendem esta fase são as seguintes:

- Seleção do ambiente de desenvolvimento dos relatórios (por exemplo: baseado na *Web*, baseado na utilização direta de ferramentas, baseado na utilização de interfaces desenvolvidas pela equipe do DW).
- Revisão e desenvolvimento de aplicações padronizadas; verificação da precisão dos dados; desenvolvimento de estruturas de navegação e documentação das aplicações de usuário final.
- Desenvolvimento de procedimentos de manutenção e atualização das aplicações de usuário final.
- Aceitação do projeto pelo usuário final.
- Revisão do projeto.

A fase de *disponibilização do DW* é composta basicamente pelas seguintes atividades:

- Plano de disponibilização do DW (por exemplo: plano de verificação da infra-estrutura, estratégia de treinamento dos usuários finais, estratégia de suporte ao usuário final, plano de atualização de versão do DW).
- Teste completo do sistema (por exemplo: executar um teste completo do processo de organização de dados, executar aplicações de usuário final, rever todos os processos).
- Disponibilização do DW propriamente dito (por exemplo: disponibilizar o DW aos usuários finais, configurar e testar a infra-estrutura, configurar privilégios de segurança).
- Aceitação do projeto pelo usuário final.
- Revisão do projeto.

Na fase de *manutenção e crescimento do DW* é composta basicamente pelas seguintes atividades:

- Contínuo suporte e treinamento dos usuários e manutenção da infra-estrutura técnica.
- Monitoramento de consultas realizadas pelos usuários finais, desempenho da organização de dados e o contínuo sucesso do DW.
- Comunicação e publicidade dos sucessos obtidos em razão do uso do DW.
- Aceitação do projeto pelo usuário final.
- Revisão do projeto.

Observa-se que este trabalho de [KIM98L] é bastante completo, pois além de abordar todas as fases, componentes e atividades de metodologias de desenvolvimento, também as discriminam na maioria das vezes de forma bastante enfática e precisa.

Outras metodologias de desenvolvimento de DW podem ser obtidas em [GAR98], [POE98] e para DM em [DYC98] .

2.9.7. Arquiteturas do *Data Warehouse*

Existem vários modelos genéricos de arquitetura de *Data Warehouse*. Estes modelos são nomeados de acordo com a quantidade de camadas que os compõe, geralmente recebem a denominação de arquitetura de uma, duas ou três camadas.

Para [GRA98], a divisão em camadas ocorre na divisão entre dados operacionais e analíticos. Já para [KIM98L], ocorre como divisão das funcionalidades de acesso ao *Data Warehouse*.

Será apresentada a arquitetura genérica de acordo com a visão de Kimball em [KIM98L].

2.9.7.1. Arquitetura de acesso aos dados

Segundo [KIM98L]:O objetivo principal do *warehouse* deveria ser gerar informação tão acessível quanto possível – para ajudar as pessoas a adquirirem a informação a qual elas precisam. E em sua visão, a arquitetura de acesso aos dados tem suas camadas nomeadas de acordo com as funcionalidades providas pelas ferramentas de acesso aos dados, disponibilizadas para os usuários finais.

Arquitetura de duas camadas – a figura 4, apresenta esta arquitetura na qual as ferramentas são projetadas para se conectarem diretamente ao componente de dados do DW.

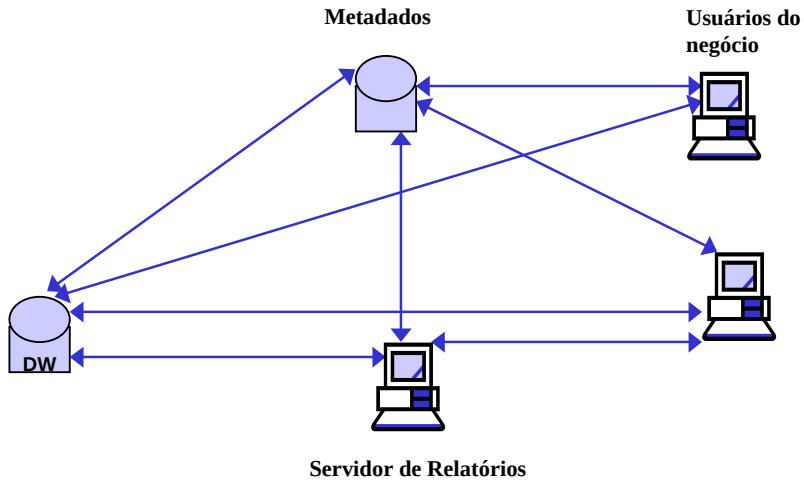

FIGURA 4 – ARQUITETURA DE DADOS DE DUAS CAMADAS

Arquitetura de três camadas (ROLAP) – A figura 5, ilustra esta arquitetura , que separa a maioria das funções de gerenciamento de consultas das ferramentas de front-end e centraliza-as em um servidor de aplicações, onde o banco de dados analítico é apresentado ao cliente como um ambiente multidimensional. As ferramentas OLAP usam intensivamente os metadados que residem em tabelas relacionais. Qualquer modificação nos componentes e estruturas do DW pode ser centralizadamente gerenciada nos metadados.

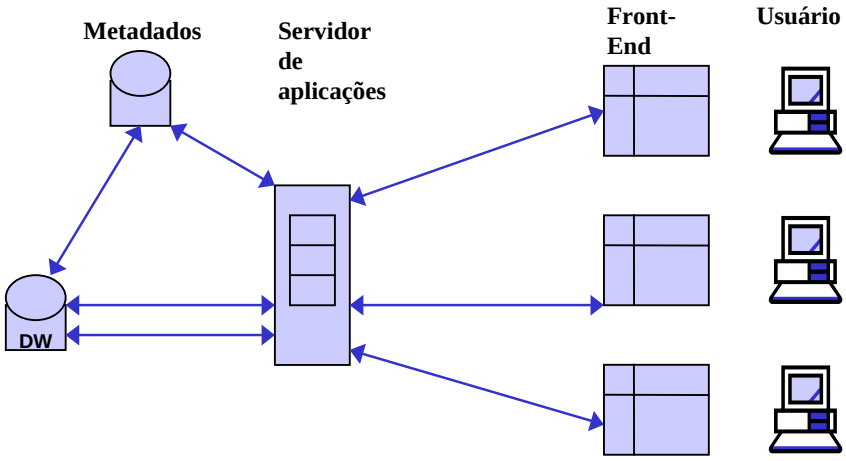

FIGURA 5 – ARQUITETURA DE DADOS DE TRÊS CAMADAS ROLAP

Arquitetura de três camadas + (MOLAP) – Esta arquitetura é ilustrada na figura 6, trata-se de uma arquitetura similar a de três camadas (ROLAP), diferenciando-se no fato de que a camada intermediária (servidor OLAP) inclui sua própria estrutura de banco de dados, denominada de banco de dados de cubo dimensional. As consultas dos usuários finais são gerenciadas pelo servidor OLAP, que as envia inicialmente ao cubo OLAP e, caso esse não possa atende-las, estas são destinadas ao banco de dados do DW. Os metadados tendem a desempenhar as mesmas funções que a arquitetura de três camadas (ROLAP).

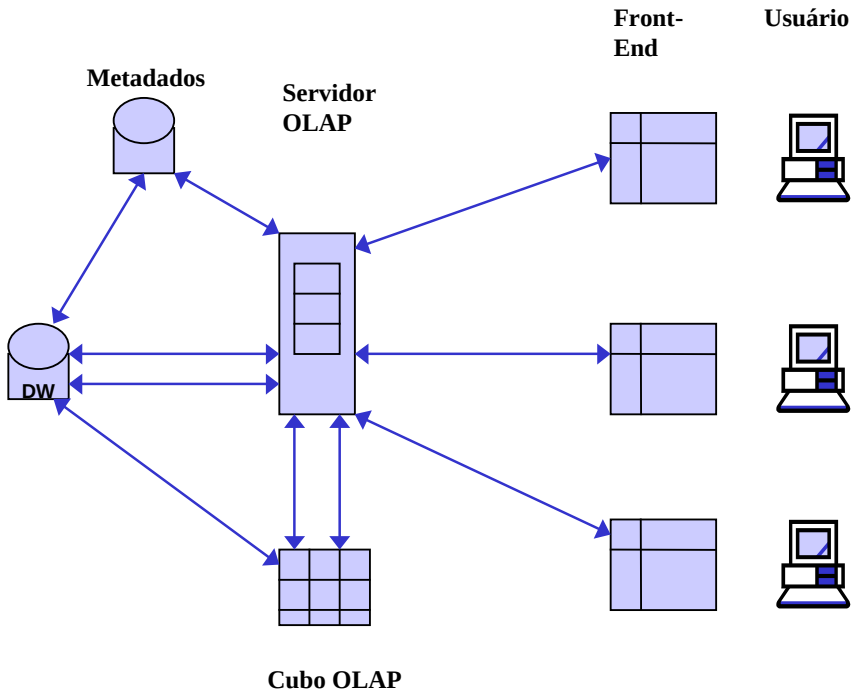

FIGURA 6 – ARQUITETURA DE DADOS TRÊS CAMADAS + MOLAP

2.9.7.2. Arquitetura Funcional

A arquitetura funcional é o plano geral de construção do *Data Warehouse*, ela descreve o fluxo de dados fontes até os usuários, bem como suas transformações e a utilização de metadados; além de especificar as ferramentas e técnicas necessárias para a ocorrência disto.

Pode-se encontrar arquiteturas funcionais de *Data Warehouse* em [INM97], [POE98], [GRA98],[GAR98], [BON98].

A figura 7, apresenta uma arquitetura funcional de alto nível, proposta por Ralph Kimball em [KIM98L]; a escolha desta arquitetura dar-se-á pelo fato de ser completa e possuir detalhes que atendem às necessidades para exposição do presente trabalho.

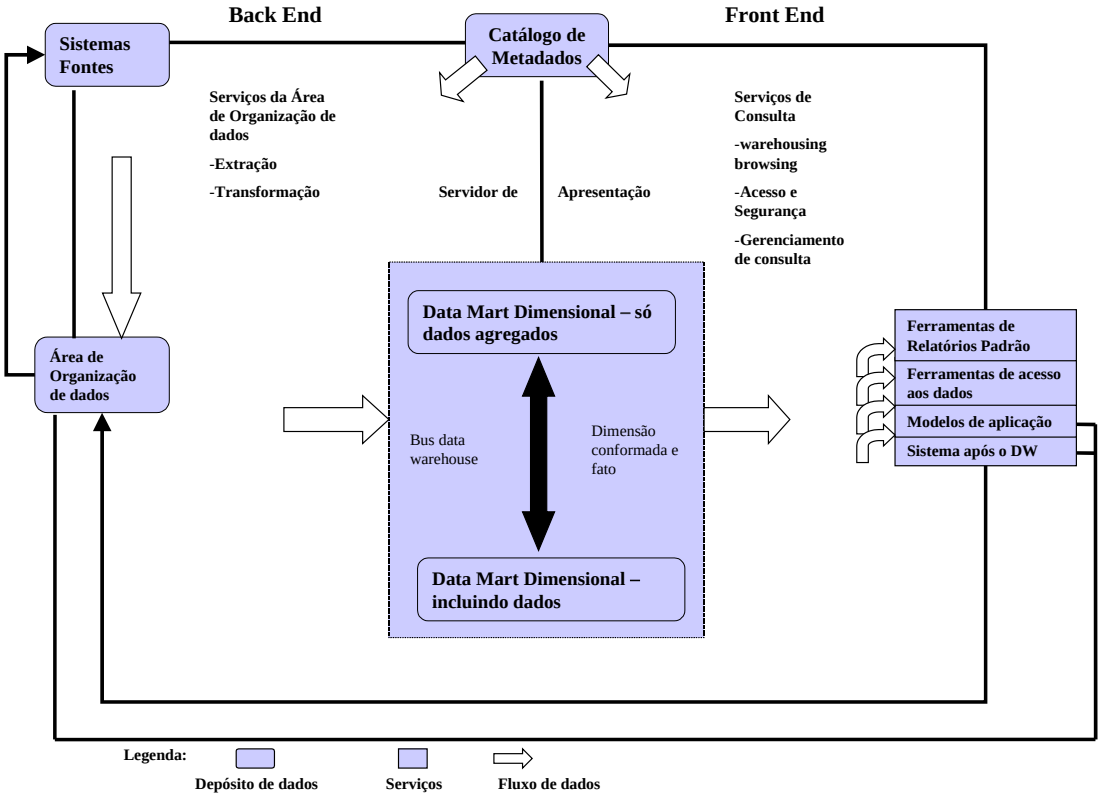

FIGURA 7 – ARQUITETURA FUNCIONAL DE ALTO NÍVEL DO DW.

Na figura 7, observa-se que a arquitetura distingue duas partes principais do *Data Warehouse*: o *Back_end* e o *Front End*. E dois componentes importantes: serviços e depósito de dados. Os depósitos de dados são lugares nos quais os dados são armazenados permanentemente ou temporariamente e os serviços são funções que possibilitam a realização de tarefas requeridas pelo *Data Warehouse*. A partir da movimentação dos dados dos sistemas fonte para a área de organização de dados, com a utilização de aplicações providas pela camada de serviços da área (processo de extração). O fluxo de dados é dirigido por metadados mantidos no cátalogo de

metadados, que descreve as localizações e definições das fontes de origem e destino dos dados; as transformações e dependências dos dados. Os dados são então combinados e alinhados na área de organização dos dados, cujos serviços dessa área (transformação, carga e controle de trabalho) são utilizados para selecionar, agregar e reestruturar dados dentro de conjuntos de dados. Os conjuntos de dados são carregados no servidor de apresentação e ligados através de fatos e dimensões conformadas que são as especificadas no barramento do *Data Warehouse* (*Data Warehouse Bus* – Anexo 2).

Após todo este processo os usuários poderão acessar os dados através de ferramentas de acesso, que geralmente, são resultantes da combinação de softwares de *front-end* criados por programadores e ferramentas disponíveis no mercado.

Back-End

O *Back-End* é a parte da arquitetura de dados onde o processo de conversão de dados ocorre.

O processo de conversão de dados pode ser conduzido utilizando-se programas desenvolvidos pela equipe do *Data Warehouse* ou através da utilização de ferramentas e técnicas disponibilizadas pela indústria. As atividades que ocorrem no *Back-End* são complexa, mesmo quando utilizadas ferramentas automatizadas, consumindo mais da metade do tempo destinado ao desenvolvimento do projeto de um *Data Warehouse*.

Os componentes do *Back-End* são: sistemas fontes, área de organização de dados e servidor de apresentação.

Sistemas Fontes – Sistemas operacionais de uma organização, aliados a fontes externas, são a base sobre a qual agem programas da camada de serviço da área da conversão de dados para obterem os dados de interesse do DW. A correta identificação dos sistemas fontes permite a escolha de ferramentas e serviços adequados para criação da arquitetura interna.

As grandes organizações adquirem informações através de dados adicionais de outras fontes de dados. Estas informações incluem demográfico, competitivo e tendências adquiridas. A meta de *data warehousing* é a liberação da informação que

está armazenada nos bancos de dados operacionais e associação destas as outras fontes de dados externas.

Área de organização dos dados – é basicamente o local de construção do *Data Warehouse*. Inclui os seguintes processos: extração, transformação, carga e indexação, verificação da garantia de qualidade, publicação e identificação de versão, atualização, consultas, auditoria, segurança e cópia de segurança e recuperação (backup e recovery).

A Área de organização dos dados ou estágio, é também denominada de gerenciamento de cópia ou gerenciamento de replicação, incluindo todos os processos necessários, tais como: selecionar, editar, resumir, combinar e carregar o *Data Warehouse* e acesso aos dados de informação de bancos de dados operacionais ou externos[ORR97].

A Área de organização de dados envolve programas de análise de qualidade de dados e filtros que identificam padrões e estrutura de dados dentro de dados operacionais existentes.

Servidor de Apresentação – São plataformas nas quais os dados do *Data Warehouse* estão organizados e armazenados para serem utilizados pelos usuários finais, e é constituído dos seguintes componentes: *Data Marts* somente com dados agregados (armazenam dados de alto nível, normalmente resumidos), *Data Marts* com dados atômicos(armazenam dados no mais baixo nível de detalhes necessários) , *Data Warehouse BUS* (barramento do *Data Warehouse*) e catálogo de metadados(descreve todo o conjunto completo de metadados usados no *Data Warehouse*, desde o passo de extração até o acesso do usuário final).

Serviços do Back-End

Os serviços do *Back-End* incluem:

- *Controle dos dados organizados*- O fluxo completo dos trabalhos de organização dos dados deve ser gerenciado através de metadados. Tendo que ser criado um ambiente para criação, gerenciamento e monitoração dos fluxos de dados, visando facilitar a manutenção e o desenvolvimento de um conjunto de estatísticas extraídas dos

metadados. Toda a funcionalidade ocorrida no processo de conversão dos dados desse ser documentada como metadados.

- *Processo de conversão dos dados* - processo que engloba os processos de extração, transformação e carga dos dados, é um dos mais complexos e que mais requer tempo de projeto do *Data Warehouse*. Para se ter uma conversação de dados de sucesso é necessário que se trace planos e estratégias de conversão e migração, para então aplicar técnicas exigidas para explorar, limpar e dar carga dos dados no *Data Warehouse*. Este processo será abordado mais detalhadamente nos próximos capítulos.

Gerenciamento de Recursos do Back-End- O *Data Warehouse* necessita manter os dados por períodos de tempo mais longo que os sistemas operacionais. E como todo sistema computacional, está sujeito a vários riscos tais como: falha ou quebra de disco, falta de energia, danos físicos a memória principal. Desta forma, deve haver projeto para processos como: backup e recovery, archive e retrieval.

- *Backup e recovery* – Permite restabelecimento do *Data Warehouse* após ocorrência de problemas. Devem prover funcionalidades, tais como: alto desempenho, administração simples e operações automatizadas.
- Archieve e retrieval – Permite o acesso aos dados movidos do *Data Warehouse* para meios de armazenamento de massa, como fita, CD-ROM e discos óticos.

A elaboração de um plano de backup e Archive deve seguir as seguintes fases: determinação de um apropriado processo de backup, como por exemplo, o que deve ser gravado na cópia de segurança, a freqüência, o horário e agendamento, quanto tempo levará; implementação do processo e prática.

Front-End

O *Front-End* é a interface do usuário com o sistema. Constitui uma camada de serviço de acesso aos dados, que se localiza entre os usuários e a informação, escondendo dos usuários as complexidades em adquirir a informação e os auxiliando a descobrir o que estão visualizando. Possui os seguintes componentes: Servidor de apresentação, ferramentas de acesso aos dados, ferramentas geradoras de relatórios, Modelos de aplicação, sistemas após o *Data Warehouse*.

Servidor de apresentação – São plataformas de destino para a área interna, onde ficam organizados e armazenados os dados que estarão disponíveis para consulta do usuário final, que poderão acessá-los através de ferramentas, e programas específicos de *front-end.*

A área de apresentação é a completa operação *front-end* para o *Data Warehouse*. Esta área está disponível o tempo todo para uso do usuário final, para que este possa acessar os dados através de várias formas, incluindo consulta *ad hoc*, relatórios, e *data mining*. Uma área de apresentação é subdividida em áreas de assunto que são chamadas *Data Mart*. Todos os modelos dimensionais em todos os *Data Mart* parecem ser um pouco semelhantes, e este suíte de modelos dimensionais tem que compartilhar as dimensões chaves do empreendimento, que são denominadas dimensões conformadas [KIM1198].

Ferramentas de acesso aos dados – Através das ferramentas podem ser realizadas consultas *ad hoc* e geração de relatórios, os quais normalmente servem para identificar anomalias, determinar padrões e tendências. Permite aumentar ou diminuir o nível de detalhes das consultas sobre as tabelas dimensão e fato através de recursos, tais como: *Drill Down*, *Drill up* ou *roll up*, *Slice*, *Dice*, *Pivot* .

Ferramentas geradoras de relatórios – utiliza *Data Warehouse* e *Data Mart* como fonte primária de dados. Geram relatórios padronizados e muitas vezes possuem internamente uma cachê ou bibliotecas de relatórios que armazenam um conjunto de relatórios pré-executados que provêem tempo de resposta e apresentação ao usuário rápido.

Modelos de aplicações – Data Mining é um exemplo primário de modelo de aplicação. É uma coleção de técnicas de análise poderosa para compreender grandes conjuntos de dados. Técnicas de *Data Mining*: *clustering*, classificação e segmentação, estimativa e predição, associação.

- *Clustering* - Emprego de métodos sobre grande massa de dados com a finalidade de recuperar estrutura com significado útil. Os resultados de uma operação de clusterização podem ser usados para produzir um sumário da base de dados ou como dados de entrada para outras técnicas, como exemplo, classificação.

- C*lassificação e Segmentação* – Usam dados existentes para criar modelos de comportamento variável. A classificação envolve descobrir regras que particionem os dados dentro de diversas classes pré-definidas. A segmentação procura determinar o particionamento dos dados em grupo que deverão ser definidos a partir dos dados e não em classes pré-definidas.

- E*stimativa e predição*- São duas atividades similares cujo resultado são geralmente medidas numéricas. Por exemplo, a estimativa da probabilidade da falência de uma empresa baseada em um conjunto de balancetes.

- A*ssociação* – Identifica eventos ou transações que ocorrem simultaneamente , determinando significativos relacionamentos entre os itens de dados armazenados. A identificação de tendências para entender e explorar padrões de comportamento dos dados é seu grande objetivo. Por exemplo, associar vendas de produto em Supermercado a compra de outros produtos; a identificação deste tipo de comportamento pode elevar as vendas, com a disponibilidade dos produtos em prateleiras próximas.

Serviços de acesso aos dados

Os serviços de acesso aos dados cobrem cinco principais tipos de atividades no *Data Warehouse*:

warehouse browsing (pesquisador de metadados ou *warehouse*) – Tem a finalidade de auxiliar usuários no acesso às informações que necessitam. Para tal, a ferramenta *browser* deve ser ligada ao catálogo de metadados.

Serviço de acesso e segurança – Para este serviço utiliza-se os serviços de autenticação e autorização, para identificar o usuário (normalmente através de senha) e permitir ou não, a utilização de recursos.

Serviço de monitoramento de atividade – Atividades de monitoramento envolve a obtenção de informações sobre o uso do *Data Warehouse*. O serviço pode ser centrado sobre as seguintes áreas: desempenho, suporte ao usuário, marketing, planejamento.

Serviço de gerenciamento de consultas – Consiste em um conjunto de funcionalidades que gerencia as modificações realizadas entre a formulação de uma consulta e sua execução no banco de dados e o retorno da resposta ao usuário. As principais funcionalidades incluídas na arquitetura são: simplificação do conteúdo, reformulação de consulta, redirecionamento de consulta e *SQL* de múltiplas partes, consciência de agregados e dirigir consultas.

Serviço de localização de consultas – Os serviços de consulta podem ser localizar em três pontos, em uma arquitetura de três camadas: no usuário final, no servidor de aplicação ou no banco de dados. Na atualidade, esses serviços, em sua maioria, estão disponibilizados em ferramentas para o usuário final.

Serviços de padronização de relatórios – Possibilita a criação de relatórios em formatos pré-definidos, possui limitada interação com o usuário. Podem ser executados em horários previamente programados. As suas funcionalidades devem incluir: ambiente de desenvolvimento de relatórios, capacidade de variação de parâmetros, execução de

relatórios baseados em tempo ou evento, flexibilidade na entrega dos relatórios, capacidade de publicar e subscrever, ligação entre relatórios, distribuição em massa, capacidade de pesquisar uma biblioteca de relatórios e ferramentas de administração do ambiente de relatórios.

Funcionalidades das Ferramentas de acesso aos dados- Os dados disponíveis no servidor de apresentação são utilizados por gerentes, chefes e administradores. Para obtenção desses dados, as ferramentas deveriam disponibilizar: ferramentas de *front-end* que gerassem relatórios complexos, controles simples de navegação, substituição automática do conteúdo de relatórios, possibilidades de interagir com várias conexões simultâneas e múltiplas fonte de dados.

As ferramentas de acesso aos dados devem incluir as seguintes características técnicas:

- Multitarefa – usuários ser capazes de executar programas e consultas enquanto uma consulta já está sendo executada.
- Cancelar uma consulta que está sendo processada sem interferir em outras.
- Conectividade com outras fontes de dados e outros bancos de dados.
- Agendamento – as ferramentas devem possuir algum recurso de agendamento, permitindo que consultas sejam processadas em qualquer horário.
- Metadados – o administrador deveria ser capaz de definir simples subconjuntos do Data Warehosue, incluindo junções pré-definidas, descrição de negócios, lista de escolha.
- Administração de software – deve ser disponibilizado utilitário que permita atualizar qualquer programa, modelo de dados, software de conectividade, a partir de uma local central.
- Segurança – a ferramenta deveria dispor de um sistema de autenticação. No ambiente do *Data Warehouse*, a segurança para ser efetiva tem que ser realizada em conjunto com o sistema de redes e banco de dados.

Esta arquitetura é uma das mais utilizadas na atualidade, embora existam outras. Tais como, a proposta por *Curt Hall* em [HALL99] que é o *exploration warehouse* (Anexo 1), que se divide em duas partes, uma delas atendendo as necessidades dos

projetistas e desenvolvedores, através da técnica de *warehouse* protótipo e a outra atendendo aos usuários finais, que é o *warehouse* de exploração que tem como objetivo facilitar o acesso aos dados.

A outra proposta é de Ralph Kimball em [KIM1198], trata-se da arquitetura *Data Warehouse* bus (Anexo 2), que é uma arquitetura constituída de vários *Data Marts* conectados a um *Data Warehouse*, via barramento do *Data Warehouse*. A concepção desta arquitetura é utilizada para atender a utilização de *Data Warehouse* na *Web*.

2.9.8. Topologias do *Data Warehouse*

Os componentes de um *Data Warehouse* podem ser descritos usando várias topologias [BON98]:

Data Warehouse Centralizado – Um *Data Warehouse* centralizado possui uma topologia simples, por constitui-se de um único *Data Warehouse* que atende a todos os clientes e as mais diversas aplicações. A escolha desta topologia para determinados empreendimentos busca obter as vantagens de economia e um sistema de gerenciamento centralizado.

Data Warehouse e Data Marts – O *Data Warehouse* é contrastado freqüentemente com *Data Mart*, que tipicamente contém um escopo de dados caracterizado por um único assunto, uma única função empresarial, ou até mesmo uma única aplicação. A característica desta topologia é a ligação de vários *Data Marts* a um *Data Warehouse*. Os usuários se conectam aos seus respectivos *Data Mart*, sendo também possível acessar diretamente o *Data Warehouse*. Esta topologia facilita a entrega de dados limpos e integrados do *Data Warehouse* para os *Data Marts* independentes.

Data Warehouse Distribuído – Esta topologia consiste de *Data Warehouses* conectados por redes com forte suporte a processamento distribuído. Os usuários conectam-se a qualquer *Data Warehouse*, em qualquer lugar, para trabalhar, como se os dados residissem em um único e centralizado *warehouse* corporativo, embora esteja distribuído fisicamente entre *Data Warehouse* múltiplos. Esta topologia tem que

suportar uma forte capacidade para gerenciamento de banco de dados distribuídos, e caso as aplicações exijam constantes operações de junções entre tabelas distribuídas, podem ser apresentados sérios problemas, podendo até chegar a inviabilizar a utilização do sistema para apresentar informações solicitadas.

Desenvolvimento estratégico híbrido – Consiste no desenvolvimento estratégico *bottom-up* de *Data Marts*, combinados com uma modelagem de dados *top-down* de alto nível. Este desenvolvimento híbrido, inicia-se com o desenvolvimento de um ou mais *Data Marts* em uma base organizada, especificada na fase de planejamento, possibilitando a criação de sistemas que sejam flexíveis e escaláveis, na medida de seu crescimento.

Segundo Kimball em [KIM98L], no futuro, um *Data Warehouse* consistirá de dezenas ou centenas de máquinas separadas com grande diversidade de sistemas operacionais e sistemas de bancos de dados, incluindo todos os tipos de ferramentas *OLAP*. O *Data Warehouse* consistirá de muitos *Data Marts* e o *Data Warehouse* global será um sistema distribuído, fundidos juntos em uma simples visão arquitetural.

2.9.9. Eliminação dos Dados do *Data Warehouse*

Os dados não povoam infinitamente um *Data Warehouse*. Eles possuem um ciclo de vida.

Há diversas maneiras pelas quais os dados são eliminados ou os detalhes são transformados, dentre os quais destacam-se [INM97]:

- Os dados são acrescentados a um arquivo de resumo rotativo onde os detalhes são perdidos.
- Os dados são transferidos de um meio de armazenamento de alta performance como o DASD para um meio de armazenamento em massa.
- Os dados são efetivamente eliminados do sistema.
- Os dados são transferidos de um nível de arquitetura para outro, como do nível operacional para o nível do *Data Warehouse*.
- Os dados são eliminados ou transformados no interior do *Data Warehouse*. Eles são simplesmente passados para níveis mais altos de sumarização.

Neste capítulo foram apresentados os principais conceitos do *Data Warehouse* e relacionados a ele. O estudo proporcionou um certo domínio bibliográfico sobre estes conceitos, estabelecendo as relações necessárias ao entendimento dos capítulos subseqüentes.

CAPÍTULO III

Migração de Dados

3.1. Introdução

Um *Data Warehouse* é povoado com dados a partir do ambiente operacional que pode possuir diversas fontes de informação espalhadas tanto geograficamente quanto por plataforma de software e hardware. O povoamento do *Data Warehouse* dar-se-á através da aquisição de dados que envolve os processos que identificam, capturam, e transformam dados em sistemas operacionais, bem como, o processo da carga dos dados que pode ocorrer em um *Data Warehouse* ou *Data Mart*. A migração de dados pode ser uma das partes mais complexas, demoradas, e de altos custos para construir e gerenciar *Data Warehouse*. O motivo principal é que empresas descobrem freqüentemente que o dados que eles querem ou carregam no armazém não existem ou são lamentavelmente inexatos e incompatíveis. As funções principais executadas em migração de dados são comumente definidas como Extração, Transformação, Transporte, e Carga de dados.

O processo de extração dos dados, envolve adquirir dados relevantes ao negócio nos bancos de dados operacionais. Estes dados extraídos são transformados (sofrem sumarizações e agregações) para que possam ser mais adequados para suporte à decisão. Em seguida, passam por processos de limpeza e de consistência das informações obtidas pelos processos anteriores. Ao término destes processos, é feita a integridade dos dados, onde ocorre a padronização de definições, taxonomia de atributos, e o registro das variações que possam ser encontradas nas múltiplas fontes de dados, tanto no ambiente operacional, como nas tendências externas.

É necessário que os atributos e dados relevantes sejam cuidadosamente tratados, para então povoarem o DW; principalmente, para tornar que seja possível aos processos de acesso aos dados uma solução eficiente.

Neste capítulo será abordado todo o processo de migração dos dados - para alguns pesquisadores este processo pode receber várias denominações tais como, aquisição dos dados, conversão dos dados, transformação dos dados, etc – que envolve os processos de extração, limpeza, transformação, carga e todos os aspectos que estes possuem.

Estará sendo abordada também, a qualidade de dados. Assim como, apresentadas algumas técnicas que pesquisadores e profissionais da área propõe para o sucesso destes processos.

3.2. Processo de migração de dados

O processo de migração de dados começa com a extração dos dados fontes (geralmente sistemas legados) para uma área de estágio comum (esquema intermediário) para que então sejam executados todos os demais processos necessários para se chegar ao povoamento do *Data Warehouse*. Mas por que os sistemas legados não podem ser a fonte de dados efetiva para o processamento de informações? Por que devem ser substituídos por *Data Warehouse*? Segundo [SQUI95], a substituição de sistemas legados por *Data Warehouse* para processamento de informação efetiva, consiste em entender a dificuldade em se obter informações a partir de velhos sistemas legados e que as razões principais em se obter tais informações consistem no fato de que os sistemas legados antigos foram projetados atendendo exigências empresariais que eram pertinentes ao negócio, às vezes a vinte e cinco anos atrás. Desta forma, estas aplicações geralmente não refletem os objetivos do negócio atual. Normalmente, as aplicações existentes nestes sistemas legados antigos, foram geradas a partir das necessidades em resultados operacionais imediatos da corporação, estando todos os dados armazenados de forma detalhada não tendo desta forma uma documentação histórica capaz de atender uma análise de visão à longo prazo dos negócios da empresa e nem a possibilidade de identificar padrões e tendências, que são feitos de maneira mais eficiente, quando os dados estão sumarizados. Associado a isto, se tem o fato das aplicações terem sido construídas uma de cada vez, e com raras integrações de dados entre elas. Tornando desta forma a perspectiva de unificar os dados em uma única base algo não muito confiável e nem tampouco fácil de realizar. Ficando desta forma a arquitetura mais apropriada a que utiliza como componente principal o *Data Warehouse*. Uma vez que, esta estrutura atende as necessidades do processamento analítico.

Nas próximas seções estará sendo analisado todo o processo que envolve a migração de dados, verificando as quatro abordagens para os passos necessários para migrar dados de sistemas fontes para o *Data Warehouse*.

3.2.1. Arquitetura de extração de dados –1ª abordagem

A primeira abordagem que será analisada sobre o processo de migração de dados é a arquitetura de extração de dados de um sistema legado para um *Data Warehouse* proposta por Ralph Kimball em [KIM696]. Esta arquitetura é composta de 13 passos para acessar todos dados legados e o modo para publicar as mudanças resultantes no *Data Warehouse* final. Estes passos compreendem basicamente duas partes do processo de migração de dados que é a extração e a carga dos dados. Como a migração de dados envolve outro importante processo que é a limpeza dos dados, será apresentado associado a esta arquitetura, o processo de limpeza dos dados, que é um processo que ocorre entre um e o outro supracitados.

3.2.1.1. Extração dos Dados

A atividade de extração dos dados fonte incluem rotinas de extração de dados que lêem os dados, converte estes dados em um esquema intermediário e move-os para uma área de estágio comum, que se trata de uma área de trabalho temporária na qual os dados são mantidos em esquemas intermediários[KIM696].

O processo de Extração dos dados subdivide-se em: Ler dados dos sistemas legados, Determinar mudanças para identificar novos dados, generalizar chave e combinar registros de fontes múltiplas[KIM696].

Ler os dados dos Sistemas Legados – A leitura dos dados dos Sistemas Legados pode ser uma tarefa muito simples ou muito complexa. Quando o Sistema de origem dos dados for uma aplicação de legado aberta e bem documentada será uma tarefa muito fácil. Mas, como normalmente não é isto o que acontece, pois na grande maioria das vezes, a origem dos dados, provém de sistemas legados sem documentação sobre o significado dos dados,e com uma estrutura interna de armazenamento de complexo

entendimento. Entretanto, existem tarefas piores, que é quando os sistemas possuem uma estrutura proprietária, onde não se sabe o formato dos arquivos subjacentes; ficando o acesso às informações restrito a relatórios ou *jobs* pré-definidos de extração.

A utilização de ferramentas e construção de programas de extração se faz necessária

Determinar mudanças para identificar novos dados – A identificação de dados que deverá ser carregado no *Data Warehouse*, reduz drasticamente a quantidade de dados que irá migrar para ele. Distinguir dados novos, de dados já lidos anteriormente no processo de extração, não é das tarefas mais simples[INM696].

Segundo [BOKU98] existem várias técnicas que estão disponíveis para atualizar um *Data Warehouse* incrementalmente, um *Data Mart* ou outro sistema operacional. São técnicas de captura de dados modificados que se enquadram em duas categorias gerais: estático e com incremento.

A captura de dados estática é normalmente associada em tomar um snapshot dos dados em um momento particular no tempo. Em alguns casos, o conjunto completo dos dados pode ser restaurado, mas provavelmente somente um subconjunto será usado.

Alguns métodos de capturas estáticas:

- captura estática - esta técnica é a mais simples das técnicas de captura de dados. O princípio fundamental é levar um snapshot do sistema operacional periodicamente e então carregar a informação no armazém de dados. A operação de carga pode acontecer em dois modos, ou completa recarga ou carga anexada. A recarga completa assume que as tabelas que são povoadas no armazém de dados são limpas, para diminuir e para que cada tabela seja recriada ou todos os dados nas tabelas anteriormente recarregadas são apagados. Esta técnica não provê um mecanismo para capturar dados históricos a menos que o sistema operacional mantenha dados periódicos. Já o modo de carga anexada assume os dados existentes nas tabelas a ser carregadas e a informação dos lugares nessas tabelas baseadas em regras predefinidas. Por exemplo, se um registro existe, é feita a sobreposição do registro inteiro.

- Captura de *timestamps* - Este método de captura de dados com incremento é semelhante, em conceito, com o método de captura de dados estático. A distinção importante no caso da abordagem de *timestamp* é que todos os registros contêm informação que pertence ao momento (tempo) ao qual eles foram atualizados por último. Estes indicadores temporais (etiquetas), conhecidos como *timestamps*, prover o critério de seleção para a captura de registros modificados. Por exemplo, a aplicação que é responsável para a recuperação de registros modificados saberá que, para as tabelas de interesse, todos os registros com *timestamps* após a última vez que o programa executou até o tempo atual serão capturados. Para tal é exigido manter um status persistente do ponto em que "fora deixado" de forma que não seja sobreposto no processo de captura com incremento. Uma vantagem para esta abordagem é que é independente de tipo de banco de dados. A segunda vantagem distinta desta abordagem em cima da abordagem de captura estática é que o volume total de dados é menor. Se o negócio não requer a captura de todas as mudanças de estado para registros de interesse, esta técnica trabalharia bem. O método de *timestamp* sofre dos mesmos problemas que a técnica de captura estática porque é difícil de capturar estados intermediários de dados a menos que, o sistema operacional fosse arquitetado para que isso ocorresse. Por exemplo, se um registro mudasse de estado cinco vezes desde sua última captura e o sistema operacional não mantivesse história periódica para o registro, então a captura de informação recuperaria só o estado atual do registro em questão. Um problema adicional com esta técnica ocorre quando da manipulação de delete. Tipicamente em um sistema operacional, registros que não são mais válidos ou usados são apagados. Para que o *timestamp* capture aplicação para descobrir a deleção, o registro deve ser marcado inativo até que seja capturado. Para só então, ser removido do sistema operacional.

Quando uma aplicação registra o momento da última alteração ou atualização em um registro, a varredura para o *Data Warehouse* pode ser executada de forma bem eficiente, porque os dados que apresentarem datas diferentes das procuradas não precisarão ser manipulados[INM97].

Idealmente todos os sistemas legados deveriam possuir um *timestamps* para seus registros modificados na base. Contudo, quando os registros modificados não são *timestamped*, a equipe de projeto do *Data Warehouse* tem que manter tabelas enormes de metadados que contém referências aos dados feitos na última extração para poder comparar os registros e identificar registros novos e alterados[INM696].

- Captura de comparação de arquivo - Este método também é conhecido como o método diferencial de snapshot (instantâneo). Este método trabalha mantendo imagens antes e depois de arquivos que diz respeito ao *Data Warehouse*. Estes registros são comparados para que sejam encontradas mudanças, e também são comparadas chaves de registro para achar inserção e deleção. Esta técnica é muito apropriada no caso de sistemas legados devido ao fato que *triggers* tipicamente não existem e *log's* de transação também não existe. Desde que a maioria de bancos de dados legados tenha algum mecanismo para acomodar dados em arquivos, esta técnica cria instantâneos periódicos e então compara os resultados para produzir registros de mudança. Esta técnica é complexa em natureza e tipicamente não desejável mas, em alguns casos, pode ser a única solução.

Excluindo o caso de captura de comparação de arquivo, as aproximações de captura estáticas são relativamente simples.

A captura de dados com incremento é um modelo dependente do tempo para capturar mudanças para sistemas operacionais. Esta técnica aplica-se melhor em circunstâncias onde mudança nos dados é significativamente menor que o tamanho dos dados fixados para um período específico de tempo (i.e., o tempo entre capturas). Estas técnicas são mais complexas que captura estática, porque elas são ligadas ao SGBD ou ao software operacional que atualizam o SGBD. Três diferentes técnicas nesta categoria são assistidas por aplicação: captura, captura baseada em *triggers* e captura de *log* de transação. Em circunstâncias onde são usados SGBD's para armazenar dados operacionais, captura de *log* de transação é a mais poderosa e provê a aproximação mais eficiente para captura com incremento.

- Captura assistida por aplicação - Este mecanismo para captura de dados não requer um processo/aplicação separado para executar a identificação de mudança e coleta. Esta técnica exige implementar a lógica de descoberta de mudança como parte da aplicação operacional. O princípio subjacente aqui é que quando são escritas mudanças no banco de dados operacional, também são escritas mudanças em uma área persistente para recuperação mais atualizada. A latência nesta técnica é obviamente mínima e significa que os registros modificados estão disponíveis para uso imediato. Embora esta técnica possa ser poderosa quando projetada e desenvolvida corretamente, sofre alguns problemas. Desde que a aplicação é responsável para fazer mudanças no banco de dados, é mais que provável manter informação chave só para registros que requerem atualizações. Então para escrever registros completos à fila de mudança, a aplicação teria que: 1) ter que ir para o banco de dados e recuperar o registro inteiro; 2) aplicar a mudança; e 3) escrever isto fora para a fila de mudança. Isto pode prejudicar o desempenho e aumentar a complexidade global. Além disso, esta técnica requer que a aplicação execute tudo necessário da computação, para atualizar registros específicos com precisão. Por exemplo, se é esperado que o banco de dados some a data atual para um registro, então a aplicação ou tem que executar aquela mesma função ou voltar e ler o registro depois que for escrito. Uma das maiores desvantagens para a abordagem de captura assistida por aplicação é que adicionar este nível de funcionalidade para sistemas legados pode ser difícil, especialmente quando os originadores do código não estão freqüentemente muito tempo dentro da organização.
- Captura baseada em *triggers* - São armazenados procedimentos *triggers* que são invocados quando certas condições ou eventos acontecem. A granularidade e flexibilidade das condições e eventos que causam um disparo para *trigger* são banco de dados específico. Esta técnica assume que o SGBD apóia *triggers*. Então, armazenamento baseado em arquivo não pode usar captura baseada em *trigger*. Esta abordagem é bem parecida à técnica assistida por aplicação na qual *triggers*, uma vez ativadas, podem executar tarefas especializadas. Neste caso, a invocação do *trigger* pode ser usada para economizar registros de interesse modificado de uma área de armazenamento persistente para recuperação. Uma

desvantagem para esta abordagem é que só deveria ser usada em casos onde o número de eventos capturados está dentro dos limites de desempenho do sistema. Por exemplo, se são capturadas todas as atualizações para uma tabela, este método cria uma carga de trabalho dupla para o banco de dados. Um problema adicional com esta abordagem é que requer que a granularidade da informação no banco de dados designado corresponda a do banco de dados fonte. Um benefício para esta abordagem é que a captura de dados acontece à fonte dos dados e, então, ambas imagens do antes e depois está disponível.

- Captura de *Log* de transação: Compreende uma abordagem ligeiramente diferente para capturar a informação modificada. Esta técnica move o *logging*(registro) e capacidades de recuperação de um SGBD e, então, arquivos seqüenciais e indexados não podem tirar proveito deste método. Como os *log's* de transação são utilizados pelo SGBD como um lugar para armazenar informação de transação para registro e recuperação, é a localização ideal para capturar informação modificada sem impedir diretamente o desempenho do banco de dados. Quando lendo os *log's* de transação, deve ser tomado cuidado particular para obter só informação que foi cometida pelo banco de dados. Esta técnica limita a quantidade de trabalho que o SGBD tem que executar, mas uma aplicação deve ser escrita para monitorar os arquivos de *log* e capturar os dados de interesse. Esta técnica é popular em algumas das técnicas de replicação de banco de dados que estão disponíveis hoje. Uma das desvantagens para esta abordagem é que depende do fato dos *log's* de transação permanecerem disponíveis até as mudanças de interesse serem capturadas. Por exemplo, se um *DBA* decide limpar o *log* de transação antes da captura dos registros mudados, a informação pode ser perdida. O método de captura de *log* provavelmente é a abordagem mais eficiente da captura com incremento pelo fato de que a técnica de *log* já esta bem aperfeiçoada e disponível na maioria das plataformas de SGBD hoje. Além de que, a captura de *log* pode ocorrer em um processador separado para reduzir o impacto de desempenho do sistema operacional.

Quando sistemas fontes não possuem registro fidedigno de data, impossibilitando reconhecer se ocorreu uma carga prévia (novas transações, atualização e deleção de registros), as novas extrações devem ser realizadas utilizando-se como base arquivos de

log de transações ou outros dispositivos que identifiquem os registros modificados [INM96].

A estratégia a ser utilizada na atualização incremental do *Data Warehouse* deve ser selecionada de acordo com a identificação das necessidades e metas do negócio.

Generalizar chaves – Uma aplicação de administração de chaves deve ser adotada.

As chaves de entrada operacionais geralmente precisam ser reestruturadas antes de serem gravadas. Muito raramente uma chave de entrada permanece inalterada ao ser lida no ambiente operacional e gravada no ambiente do *Data Warehouse*. Em casos simples, um elemento de tempo é acrescentado à estrutura de chave. Em casos complexos, toda a chave de entrada precisa passar por um novo processo de *hashing*, ou ser reestruturada.

Combinar registros de Diversas Fontes - Na grande maioria dos *Data Warehouses*, os dados provém de vários Sistemas Fontes diferentes e independentes. O estabelecimento de um ambiente intermediário de armazenamento de dados se faz necessário. Para tal, um trabalho de desnormalização dos dados das dimensões deve ser feito para que possa aproximá-los do esquema final que será carregado no *Data Warehouse*.

3.2.1.2. Limpeza dos dados

A limpeza dos dados é um dos processos mais importantes no processo de migração dos dados para o *Data Warehouse*. Com a evolução e construção de *Data Warehouse* ao longo dos anos; alguns importantes questionamentos começaram a ser feitos; questionamentos estes que visam cada vez mais a busca pela qualidade de dados. Alguns questionamentos mais comuns foram elaborados e respondidos por [LARR99], em seu artigo "*Data Cleansing in the Data Warehouse"*. Os questionamentos são os seguintes: Onde se deve limpar os dados? Quando deixar de limpar dados? Onde está o limite entre limpeza automatizada e humana? Como lidar com dados que possuem identificadores múltiplos? Como administrar limpeza de dados externos?, que devem ser respondidas para que se tenha uma limpeza de dados próspera.

Para a pergunta que é sobre o local *aonde deve ser executada a limpeza dos dados*, [LARR99] afirma que se os dados utilizados são do banco de dados fonte da corporação; então, os dados devem ser limpos na própria fonte para então ser transformados ao *Data Warehouse*. Pois, caso contrário a empresa poderá ter problemas tais como:

- Dados defeituosos permanecerão no banco de dados fonte, com a limpeza dos dados feita fora da fonte, e quando processos forem executados usando estes dados defeituosos, estes processos falharão e continuará incorrendo custos empresariais do fracasso de processo e custos sobre informação fragmentada e refeita.
- Os dados fontes não corrigidos terão o potencial para corromper o *Data Warehouse* à medida que dados modificados com o passar do tempo são propagados da fonte.
- Relatórios dos dados fonte e dos dados limpos do *Data Warehouse* que deveriam ser equivalentes, não serão e causarão confusão e falta de confiança.

Para [INMON99] limpar e testar dados no ambiente de sistema legado provê absolutamente a melhor fundação para todos os tipos de processos, inclusive *Data Warehouse*. Mas, se o ambiente for muito antigo e sem integração, o tempo e quantidade de trabalho exigidos para limpar os dados certamente afetará o tempo de entrega do *Data Warehouse*, o que é uma grande desvantagem.

No caso dos dados não serem limpos na fonte, deve ser avaliado a sua qualidade (perfeição e precisão), com o propósito de determinar sua confiabilidade e a necessidade para melhoria de qualidade de informação [LARR99]. A avaliação dos dados compreende o grau de obediência dos dados às regras de negócio, isto é, um valor válido, dentro do limite certo, relacionado a um objeto referenciado válido (Validez) e a verificação do valor válido para avaliar se é um valor correto (Precisão). Por exemplo, dados freqüentemente possuem uma alta incidência de valores válidos, especialmente valores omissos que são válidos, mas não corretos.

Na limpeza fora do ambiente de origem, é necessário que todos os processos e consultas sejam direcionados para os dados corrigidos, que devem ser os novos registros de referência.

A qualidade (perfeição, validez e precisão) dos dados do *Data Warehouse* deve ser dimensionada e informada, para que os tomadores de decisões possam avaliar a confiabilidade dos dados, para só então decidirem quais medidas adotar.

Embora a limpeza dos dados seja importante e necessária, tanto [INM99] quanto [LARR99], concordam que desde que se saiba a extensão, pode haver alguns graus de imperfeição de dados no armazém de dados.

A outra importante questão sobre limpeza de dados é saber *quando deixar dados sem limpeza*. Os sistemas legados possuem muitos dados sujos, mas pelo fato de não ter sua funcionalidade afetada por causa disto, nunca houve a preocupação da limpeza desses dados. Logo, normalmente dados que irão migrar para o ambiente de *Data Warehouse* necessitam de correção e isto requer uma avaliação de qualidade destes dados. Para [LARR99] esta avaliação pode ser feita da seguinte forma:

- Dados sem avaliação devem ser rotulados como “dados não auditados”, isto indicará que o nível de confiabilidade não está assegurado.
- Atributos no *Data Warehouse* podem ser priorizados como A,B e C, onde: Dados com prioridade A, são dados mortais, podem ter custo alto de fracasso. Desse modo, devem ser todos limpos. Já os dados com prioridade B, são a segunda prioridade de dados importantes e deve ser enviado como recursos permitidos e orientados por uma informação dos desenvolvedores. E finalmente os dados com prioridade C, são dados opcionais ou não criticados onde o custo de omissão e erro é marginal.

A questão da *aplicabilidade da limpeza automatizada e a humana*. Para se alcançar validez de dados podemos utilizar correção de dados automatizada. Desta forma, se pode testar eletronicamente se um endereço é válido. Alguns softwares de limpeza de endereço podem aplicar correções de endereços e até mesmo aplicar dados de mudança de endereço de serviços postais. Mas pessoas mudam-se sem remeter informação. E neste caso, se requer confirmação humana para verificar se um endereço para uma pessoa específica está correto. Para dimensionar, confirmar e corrigir a maioria dos dados para valores precisos a correção humana é utilizada. Por exemplo, em um cadastro de clientes podem ser requeridas técnicas como pesquisas de correio ou contatos telefônicos para assegurar e corrigir dados pessoais que são importantes à organização. Desta forma, a determinação do tipo de limpeza a ser utilizada não pode ser previamente determinada, pois ambas podem ter aplicabilidade, tudo vai depender

dados que estarão sendo analisados.

Um dos mais árduos problemas de limpeza de dados é *lidar com dados que tem identificadores múltiplos*. Quando muitos arquivos contem muitos registros redundantes sobre uma única entidade, e esses possuem significado embutido ou chaves não sincronicamente definidas, se deve buscar consolidar estes registros duplicados dentro de um único arquivo ou banco de dados. Para tal, se deve manter uma tabela de referência cruzada para relacionar a ocorrência de registro para os registros que previamente existiram, mas não mais executam. Isto é usado para redirecionar qualquer transação empresarial que usa velhos identificadores para a ocorrência de registro. Também se deve manter um arquivo de auditoria com imagens dos dados de antes e de depois para poder assegurar a reconstrução dos registros originais no caso de uma desconsolidação. Os arquivos estando com uma única ocorrência de registro podem ser comparados e consolidados através do cruzamento de arquivos redundantes que selecionam os valores mais fidedignos por propagação para o *Data Warehouse*. Os valores dos dados de cada fonte devem ser corrigidos e sincronizados por consistência para uma possível extensão. A equivalência dos dados deve ser mantida nos arquivos redundantes, assim como, a tabela de referência cruzada das ocorrências relatadas.

Uma outra questão de limpeza de dados é *como administrar a limpeza de dados externos*. Quando se trata de dados externos comprados (demográfico, perfil, censo, geoespacial, financeiro) para expandir o *Data Warehouse*, é essencial que uma garantia escrita da confiabilidade dos dados seja fornecida pelo provedor da informação. Os dados comprados devem ser sempre analisados, para que se possa saber o nível de confiabilidade. Quando da aquisição de dados externos, um dos problemas pode ser os recursos internos e perícias que podem ser insuficientes para os integrar. Além do fato de que, nesta situação é mas difícil obter informações adicionais de clientes para dirimir dúvidas quanto a estrutura dos dados.

A limpeza dos dados ocorre sobre dados sujos que estão nos sistemas legados. Mas se esse dados são tão danosos, como podem estar presentes nestes sistemas? A resposta mais comum é que são dados que não afetam a funcionalidade destes sistemas, e por este motivo, ninguém perde tempo e dinheiro tentando fazer a limpeza.

Os dados para serem corretamente limpos devem ser identificados e devem ser traçadas estratégias de limpezas para cada tipo de sujeira encontrada. Segundo [MOSS98], os

tipos mais comuns de sujeira são geradas ainda na fase de entrada de dados, onde o preenchimento de alguns campos é feito de acordo com a importância dos dados para os diversos setores da empresa. Podendo dessa forma ser preenchidos somente para passar pela crítica de campos do programa, sem está com valor correto.

3.2.1.3. Tipos de sujeiras mais comuns presente nos sistemas legados

Segundo Larissa Moss, em [MOSS98], os tipos de sujeiras mais comuns nos sistemas legados são:

Valores *Dummy* – são os valores que normalmente o usuário necessita preencher na aplicação de entrada de dados mesmo sem saber o conteúdo da informação. A preocupação é só com o tipo de dado e um valor qualquer preenche o campo, para que passe na crítica de validação. Em uma base de cliente os campos mais prováveis de receber estes valores são número de seguro social, idade de cliente, Cep, CPF, Inscrição Estadual. Para campos preenchidos com valores 000'0s e 999'9s. A identificação é simples e a limpeza pode ser executada . Mas, existem alguns casos que a pessoa responsável pela entrada de dados pode utilizar por exemplo o próprio número de registro de eleitor para preencher os dados referentes a este campo. Este tipo de identificação de inconsistência é inacessível a equipe de projeto do *Data Warehouse*.

Ausência de Dados - Campos de um arquivo ou tabela podem ter dados ausentes. Um dos motivos para isto, é fato de que departamentos de uma mesma empresa possuam necessidades diferentes para existência de certos dados em sua operação. Desta forma, o preenchimento do campo pode ser exigido ou não. Por exemplo, um departamento pode exigir a captura de informações que para outro Departamento são totalmente dispensáveis.

Campos com propósitos múltiplos - Um campo de dado pode ser usado para propósitos múltiplos depende do departamento que fez a entrada do dado e da referência cruzada em relação a outros campos. No primeiro caso, o Departamento com o propósito de atender as necessidades, redefine diversas vezes a mesma área de dados. No segundo caso, o campo a ser observado vai ser preenchido de acordo com a funcionalidade da

operação de negócios envolvida.

Dados Inconsistentes – Informações inseridas nos sistemas podem estar com conteúdo inconsistente. Por exemplo, em uma base de dados de clientes, a informação da cidade que a pessoa mora, não combina com a informação do estado correspondente, isto é, no campo cidade a informação é Florianópolis e no campo estado o conteúdo é Acre.

Uso impróprio de campos de Endereço – A falta de estabelecimento de padrões para preenchimento de campos de endereços, gerava diversos maneiras de preenchimento da mesma informação. Isto porque os campos de endereços eram linhas de textos livres a serem preenchidas a gosto do usuário. Um mesmo endereço poderia ter diversos preenchimentos, por exemplo:
Endereço: Av. Rio Branco, no. 1000, bloco B, sala 10, Centro
Endereço: Avenida R. Branco, 1000/B-10, Centro
Endereço: Avenida Rio Branco, no. 1000, bloco B, Sala 10
Endereço: Av. R. Branco, no. 1000, B-sala 10

Chaves Primárias reutilizadas - Este é um dos assuntos mais críticos sobre "dados sujos" que podem ocorrer nos sistemas legados. Os sistemas operacionais raramente armazenam história além de 90 ou 180 dias, o que faz com que freqüentemente valores de chaves primárias sejam reutilizados. Esta situação gerará um problema enorme de integração de dados em um ambiente de *Data Warehouse*.

Identificadores não únicos - A inclusão de vários registros para a mesma entidade dentro de um mesmo sistema, trará um grande problema de integração de dados, uma vez que dados que deveriam estar relacionados não estarão. Por exemplo, um cliente identificado por vários códigos de cliente diferentes.

Os dados a serem extraídos dos Sistemas legados devem sofrer uma análise, com o propósito de tentar identificar todos estes tipos de sujeiras descritas. E para cada tipo de sujeira, regras devem ser especificadas e documentadas como metadados. O processamento de limpeza dos dados, será feito com o auxílio dessas regras de

transformações (metadados) definidas para cada caso.

3.2.1.4. Carga dos Dados

Após a transformação dos dados é realizado o processo de carga, onde os dados são colocados no servidor de apresentação. Algumas funcionalidades necessárias durante o processo de carga são:

Suporte para destinos múltiplos – o destino dos dados poderá ser para *dart mart* atômico ou *Data Mart* agregado, tendo cada alvo dos dados seus próprios detalhes e sintaxe. Ficando o processo de carga responsável por reconhecer as diferenças para utiliza-las ou evita-las da maneira mais apropriada.

Otimização do processo de carga – Técnicas de banco de dados para otimizar o processo de carga para evitar a geração de *log* durante o processo, criar índices e agregar dados, podem ser invocadas dos bancos de dados ou registradas em scripts através da utilização de ferramentas sobre a área de organização de dados.

Suporte completo ao processo de carga - A eliminação e recriação de índices e particionamento físico de tabelas e índices, são algumas exigências antes e depois da carga atual, que o serviço de carga precisa suportar.
O processo de Carga, pode ser subdividido em: Criar imagem dos registros, criar agregações, generalizar chaves para registros de agregações, carregar registros com integridade referencial [KIM696].

Criar imagens de registro de carga - Neste passo verifica-se se os registros transformados estão compatíveis com os do *Data Warehouse*. Todos os registros de carga devem ter a mesma quantidade de campos que seus respectivos registros no *Data Warehouse*.

Criar agregações - Os registros agregados devem ser criados antes da carga.Criar agregações é executar várias vezes, um utilitário de seleção (*sort*) classificando cada vez

pelos critérios desejados e criando registros de quebras com o somatório dos dados. Em um *RDBMS*, essa atividade é realizada registro a registro, sendo mais eficiente quando utilizado um utilitário de ordenação.

Generalizar chaves para registros de agregação - Os registros agregados precisam de chaves. As chaves para registros de agregado precisam ser geradas de maneira completamente artificiais e não devem estar em conflito com chaves para registros de nível básico. A equipe do *Data Warehouse* precisa construir uma aplicação para gerar e administrar destas chaves de registros agregados.

Carga de registros com integridade referencial - O processo de atualização dos índices deve nesta fase ser inibido para se obter mais performance.
É fundamental a carga dos dados com integridade referencial nesta etapa, pois é o momento de identificar alguma inconsistência de dados que resta. Pois depois de carregado o *Data Warehouse* possuirá bilhões de registros, e encontrar inconsistências será praticamente impossível, além de ter custos onerosos para a corporação.

Tratar registros rejeitados - Invariavelmente existirão alguns registros que fracassam ao processo de carga. Normalmente eles fracassam por falha na integridade referencial. Cada um destes registros deve ser analisado e seus componentes ruins corrigidos. Uma carga deverá ser processada somente para estes registros.

Construir índices – Estando os dados todos carregados, todos os índices afetados devem ser reconstruídos. Uma aplicação para apoiar a observação do status destes índices se faz necessária.

Assegurar qualidade dos dados – A qualidade dos dados recentemente carregados deve ser assegurada. Para tal, uma avaliação através de uma série de relatórios e gráficos que devem ser gerados a partir do *Data Warehouse* atualizado; deve assegurar que as principais informações das dimensões encontram-se dentro dos limites esperados, bem como, os valores do *Data Warehouse* estão válidos em relação com os seus dados de origem nos sistemas legados.

Publicar nova carga realizada - . A formalização da liberação de nova versão do *Data Warehouse* atualizado, pode ser feita via manual ou talvez até mesmo uma mensagem de e-mail automatizada para todos os usuários, resumindo o estado da carga do dia prévio. Este é um mesmo passo muito apreciado pelos usuários.

Nem todos os *Data Warehouses* são construídos em uma base relacional. Alguns ambientes de *Data Warehouse* são implementados através de ferramentas *OLAP*, cuja estrutura interna de armazenamento de dados é proprietária e multidimensional. Nestes casos, os passos de criação de registros agregados e chaves de registros agregados ficam por conta da ferramenta e totalmente transparente para a equipe de projeto.

3.2.2. Plano de Conversão de dados - 2ª abordagem

A segunda abordagem é feita [BOHN97] e envolve os seguintes passos para a migração de dados: *elaboração do plano de conversão*, as *especificações deste plano*, que envolve os processos de extração, transformação, carga dos dados, e qualidade dos dados.

Segundo [BOHN97]: um fator de sucesso para conversão de dados é que todos os membros da equipe do projeto do *Data Warehouse* entendam o fluxo e requerimentos da conversão. A conversão de dados para um armazém normalmente é muito grande.

A elaboração de um plano de conversão determina a melhor rota para migrar dados fonte ao *Data Warehouse*. O plano de conversão deve considerar recursos disponíveis para o projeto, volume de extração de cada fonte de dados, as melhores formas de extração através da avaliação dos esquemas físicos, linguagens de programação apropriadas e métodos de acesso recomendados.

O transporte dos dados para a área de estágio comum tem que levar em conta os recursos disponíveis da máquina, o grau de conhecimento de conversão de dados da equipe envolvida no processo e o volume da fonte de dados. Por exemplo, no caso do sistema fonte ser um *mainframe* com *MVS* e com grande volume de dados, e a equipe técnica para conversão de dados deve possuir conhecimento em *MVS* e *COBOL*. O plano de conversão de dados deve recomendar que a área de estágio comum se localize em uma das máquinas *MVS*. Mas, caso os recursos das máquinas *MVS* sejam limitados e

o conhecimento da equipe técnica de conversão de dados seja *UNIX* e *C*, o plano de conversão deve recomendar que a área de estágio comum se localize em uma máquina *UNIX* (possivelmente o servidor do *Data Warehouse*).

O plano de conversão de dados deve também considerar a estrutura do *Data Warehouse* e dos esquemas do banco de dados destino.

Para sistemas fontes localizados em diversas máquinas, os dados deverão migrar para uma área de estágio comum. Esta área de estágio comum é um esquema intermediário, que não se trata nem do esquema de origem dos dados e nem do esquema final do *Data Warehouse*. O esquema intermediário é a interface comum para a qual todos os sistemas fontes são extraídos. Normalmente, o esquema intermediário fica entre as duas formas de esquemas (origem e final) e contém campos adicionais, como números percentuais para uso em cálculos ou campos chave para leitura de tabelas de referência, que facilitam as rotinas de limpeza, transformação e integração dos dados.

O plano de conversão de dados deverá considerar todas as fases do fluxo de dados ao longo da conversão e estabelecer um plano para os seguintes assuntos:

Como os dados migrarão dos sistemas legados para a área comum.

Como a equipe deverá limpar, transformar e integrar os dados na área de estágio comum.

Como a equipe administrará as chaves primárias e chaves estrangeiras para o *Data Warehouse*.

Como os dados serão migrados da área de estágio comum para o servidor de *Data Warehouse*.

Como os metadados serão gerados, armazenados, atualizados e exportados para o repositório de metadados do *Data Warehouse*.

Como a equipe carregará e indexará os dados no *Data Warehouse*.

Como a equipe assegurará a qualidade dos dados convertidos.

Estabelecido o plano supracitado, especificações de conversão devem ser criadas para que a conversão dos dados tenha sucesso garantido.

3.2.2.1. Especificações de Conversão

Com o plano de migração estabelecido, o próximo passo será a criação das

especificações de conversão. Para tal, será preciso analisar os dados da fonte cuidadosamente. O mapeamento de dados da fonte em relação ao banco designado deve ser feito, bem como as regras de transformação a que os dados serão submetidos.

Tipicamente, estas especificações de conversão serão documentadas em uma planilha ou um processador de textos e depois impressas para a revisão e aprovação dos usuários. Mas, existem algumas ferramentas de migração de dados, nas quais as especificações de conversão podem ser diretamente documentadas, estas ferramentas também geram relatórios necessários a esta atividade.

As especificações de conversão geradas são metadados extremamente importantes para o *Data Warehouse*. O plano de conversão de dados especifica o formato dos metadados de conversão, assim como, o armazenamento, atualização, e exportação do metadado para o ambiente do *Data Warehouse*.

Independente da forma de conversão ser automática ou manual, e das especificações de conversão em código de programação, deve-se utilizá-las para entender o mapeamento dos dados e o processamento lógico necessário. O código de programação gerado consiste em seis tipos de rotinas que efetuam as seguintes funções:

- Extração dos dados dos sistemas fonte para o esquema intermediário;
- Conversão dos esquemas intermediários para dados de carga;
- Agregação dos dados de carga;
- Migração dos dados de carga da área de estágio comum para o servidor do *Data Warehouse* (se a área de estágio comum não estiver no mesmo servidor);
- Carga dos dados no servidor de banco de dados do *Data Warehouse* em modo "*Bulk Load*";
- Validação dos dados.

Extração dos dados fontes para esquemas intermediários - O processo de aquisição de dados em uma área de estágio comum aumenta a reusabilidade do código de programação. O processo de extração de dados é projetado para isolar somente os dados necessários que devem migrar para o *Data Warehouse*. Este projeto serve tanto para a carga inicial quanto para a carga incremental (atualizações) feita no *Data Warehouse*. Sendo importante por reduzir o volume de dados que migram ao *Data Warehouse*,

trazendo com isso redução de requisitos de recursos de computador e de rede para a conversão. O projeto de extração de dados deve considerar como o sistema fonte assinala modificações e dados novos, tais como *timestamps* ou arquivos periódicos.

Timestamps e arquivos periódicos (como arquivos de fim de mês de reivindicações liquidadas em uma empresa de seguros) são ideais visto que as rotinas de extração de dados identificam facilmente e isolam os dados novos e os modificados. Se o sistema fonte não indicar os dados novos e modificados, deve-se comparar os dados fonte com os arquivos principais do *Data Warehouse* para encontrar os dados novos ou modificados.

As rotinas de extração de dados normalmente rodam dentro do ambiente dos sistemas fonte,executando funções que transformam, convertem de binário, de decimal empacotado, compara, combina e analisa os dados fonte. Estes tipos de condicionamento de dados são mais facilmente executados no seu ambiente de origem do que no ambiente de destino que pode não suportar estas funções ou tipos de dados. O processo de condicionamento de dados tem base total no conhecimento da fonte de dados, ou seja: na estrutura dos dados como residem nos sistemas operacionais ou no *Data Warehouse*; na estrutura dos dados como percebidos pelo usuário final ou pelas aplicações de *warehouse*; no conhecimento das regras exigidas para identificar, mapear, limpar, transformar, e agregar os dados; e nas características de segurança associadas com vários objetos de metadados, como *logins* de sistemas fontes e regras de segurança do banco de dados do *Data Warehouse*.

Conversão dos esquemas intermediários para dados de carga - Uma vez que os dados fontes estão agrupados em uma área de estágio comum, a execução das rotinas de conversão que devem limpar os dados deve ser iniciada. A limpeza de dados assegura a integridade dos dados por intermédio de programas especiais que corrigem os dados, melhorando a precisão dos dados e utilidade global. São utilizados softwares que permitem a correção e melhora de campos de dados como nomes e endereços.

O projeto da arquitetura do *Data Warehouse* deve acomodar restrições impostas pelo software de limpeza de dados (em quais plataformas o software de limpeza de dados pode ser executado ou não), para que a equipe de conversão de dados consiga implementar os serviços de limpeza de dados. Este tipo de software somente deve ser

usado caso não exista possibilidade de correção dos dados na sua fonte. Além disso, todos os requerimentos de limpeza devem ser identificados no plano de conversão.

Os componentes principais de limpeza de dados são:

- O exame minucioso dos dados que determina a qualidade dos dados, os padrões existentes dentro deles e a cardinalidade dos campos (o número de campos diferentes usados);
- Análise de dados determina a composição e o destino de cada componente de cada campo;
- A correção dos dados compara os dados contra lista conhecida (normalmente endereços), garantindo que todos os campos sejam marcados como "bom", "ruim" ou "corrigido automaticamente". Sempre que possível a equipe de conversão deve trabalhar em conjunto com o cliente para efetuar a correção dos dados fonte;
- A comparação de registros determina quando dois registros (talvez de tipos diferentes) representam dados do mesmo objeto. Este processo envolve muitos julgamentos e requer uso de ferramentas sofisticadas.

Outras rotinas de conversão transformam os dados. Normalmente, os sistemas fonte possuem atributos de dados que necessitam de um processo de transformação. Por exemplo, os códigos do sistema fonte para identificação do sexo de uma pessoa são "1" para masculino e "2" para feminino. Porém o armazém de dados codifica estes como “M“ e “F” respectivamente. O processo de transformação garante um mapeamento consistente dos códigos e chaves entre os sistemas fonte e o *Data Warehouse*.

Se a extração de dados é feita em mais de uma área de negócio ou em mais de uma versão dentro de uma mesma área de negócio, os dados devem ser integrados em uma única visão. E as anomalias de dados e correções devem ser mapeadas, a fim de serem feitas antes da migração dos dados e carga dos dados .

As técnicas de condicionamento, limpeza, transformação, e integração dos dados devem ser aplicadas de uma maneira interativa. E com a aprovação do usuário para o condicionamento dos dados, devem ser criadas imagens de registro de carga para dados de nível atômico, de dimensão, e de fato.

Dado de nível atômico é o mais baixo nível de detalhe dos dados no *Data Warehouse*, sendo preciso fornecer o nível de granularidade necessário para condicionar, limpar, transformar e integrar efetivamente os dados. Se o *Data Warehouse* alimenta *Data Marts* que são, normalmente, projetados por um esquema estrela, a carga de dados de nível atômico será usada para criar as tabelas de dimensão do *Data Mart*. Se o processamento desta atividade for seqüencial, deve-se agrupar os arquivos resultantes dos dados de carga de dimensão para eliminação de registros duplicados.

Para mapear e gerar as chaves para cada uma das tabelas dimensão de um esquema estrela, deve-se criar uma aplicação de administração de chaves. Existe uma série de estratégias para administração e geração de chaves . São exemplos, a integração de chaves de sistemas fonte e chaves geradas por sistema. Integração de chaves de sistemas fonte transformam chaves lógicas de diversos sistemas fonte em uma única chave física. Por exemplo, um paciente de um hospital pode ser identificado pelo número de código e número do plano de saúde em um sistema e pelo nome e data de nascimento em outro sistema. No *Data Warehouse* , as chaves são integradas. As Chaves geradas por sistema são aquelas atribuídas pelos sistemas de conversão e pelos sistemas de gerenciamento de banco de dados relacionais (SGBDR).

Para criar dados para carregar nas tabelas fato, deve ser gerada uma aplicação que primeiro compara os dados carregados nas dimensões com os dados de carga de nível atômico. Para então, povoar os registros de dados de fato com as chaves das dimensões e os fatos quantitativos associados contidos nos dados de nível atômico. Esta aplicação informa quais registros dos dados carregados nas dimensões que não foram comparados com a carga de dados de nível atômico. Quando a aplicação executa corretamente, não existirá absolutamente nenhuma possibilidade de *mismatch* acontecer.

Carga de dados agregados - A carga de dados é agregada executando-se uma série de tipos e aplicação de gerenciamento de chave várias vezes. O resultado é a carga de dados de todas as dimensões e a carga de dados de todos os fatos associados agregados definidos no projeto do *Data Warehouse*.

Geralmente, a execução dessas funções é feita em seqüência ao invés de dentro do SGBDR - Sistema Gerenciamento de Banco de Dados Relacionais por três razões:

- Utilitários de ordenação externos são mais rápidos;

- Os dados novos agregados são armazenados fora do servidor de *Data Warehouse* como um requisito para recuperação de falhas. Na ocorrência de falha, a recuperação dos dados é feita com a carga somente dos dados agregados;
- Um programa chamado "*bulk load*" é utilizado para povoar o banco de dados designado em lugar de uma aplicação com *S.Q.L*. embutido, por ser um método rápido e eficiente.

Transporte, Carga e indexação dos dados – Caso a área de estágio comum não esteja no mesmo servidor do *Data Warehouse*, é necessário que seja feito o transporte dos dados para este servidor. Estando os dados reunidos no servidor do *Data Warehouse*, utiliza-se um tipo de utilitário para carga de grande volume de dados em sistemas de banco de dados denominado "*Bulk Load*". Deve-se utilizar o modo de integridade referencial ligado durante este processo de carga para garantir que as chaves da tabela fato sejam verdadeiras em relação as chaves estrangeiras das tabelas dimensão. Normalmente, as tabelas fato contém milhões — talvez bilhões ou mais — de linhas. Ou seja, é bem provável que existam linhas que violem a integridade referencial, tornando fácil sua descoberta. Se não utilizar o modo de integridade referencial ligado durante este processo de carga haverá pouca chance de conseguir tomar ciência do problema.

No caso de decidir que índices devem ser atualizados durante o processo de carga ou deixar esta atualização para depois depende das restrições de tempo ou de capacidade do SGBDR. Vários SGBDRs permitem ao *DBA* segmentar a tabela de índices, desse modo pode-se remover uma parte do índice, disparar o processo de carga e reconstruir os índices logo que seja necessário. Tabelas de índices segmentadas reduzem a quantidade de tempo necessária para carregar e indexar os dados. Esta redução de tempo afeta diretamente e positivamente na disponibilidade do *Data Warehouse* e é especialmente importante se os usuários do *Data Warehouse* estão localizados ao redor do mundo com diversos fusos horários.

Assegurar qualidade de dados – A qualidade de dados é assegurada ao longo do processo de conversão dos dados. O plano de conversão especifica revisões do cliente ou do usuário final e procedimentos de aprovação , validação de dados e de correção, e o processo de reconciliação de dados com o sistema de fonte.

A criação do plano de conversão e as especificações de conversão devem ser feitas juntamente com o cliente e o usuário final. Examinar o ambiente de processamento atual para planejar o melhor meio de mover os dados dos sistemas fonte para o armazém de dados. A documentação e revisão do plano devem ser feitas com o cliente. Desta maneira, o cliente sabe o que esperar durante todo o processo de conversão dos dados e dar ciência formalizada do plano de conversão. O ideal é que o cliente seja envolvido em todo o processo de criação de especificações de conversões efetivas, sempre aprovando e formalizando ciência.

O primeiro lugar para reconciliar o processo de conversão é após a extração dos dados fonte para o esquema intermédio. Totalize o número de registros extraídos e outros dados aditivos, como contas e quantias monetárias, do esquema intermediário. Estes estão amarrados aos totais do sistema fonte. Como uma regra, os dados devem se reconciliados o mais cedo possível e freqüentemente durante o processo de conversão de dados, permitindo desta forma descobrir problemas o quanto antes para evitar ter que executar novamente vários passos do processo.

Durante o condicionamento, limpeza, transformação, e integração dos dados, decisões devem ser tomadas juntamente com o cliente dos valores atuais dos dados que estão sendo examinados. Quando descobertos dados errantes, eles devem ser mostrados ao cliente. Para que em conjunto se decida a correção dos dados no sistema fonte ou com as rotinas de conversão de dados. Caso a correção dos dados seja feita com as rotinas de conversão de dados, as correções no metadados de conversão de dados também devem ser gravadas.

O outro lugar para reconciliar os dados é após o término da carga de dados de nível atômico. A maneira é idêntica ao que foi feito no processo anterior de extração de dados. A totalização do número de registros extraídos e de outros dados aditivos deve ser efetuada. Quando os totais são diferentes deve haver reconciliação das diferenças entre os sistemas fontes e com a carga de dados de nível atômico.

Finalmente, ocorre a reconciliação dos dados após a criação e a agregação da carga de dados fato. Os totais de todas as tabelas fato batem exatamente um com os outros e com a carga de dados de nível atômico.

As complexidades do processo de conversão de dados modifica de acordo com a o projeto e implementação do *Data Warehouse*, estruturas de sistema fontes, limpeza de

dados fonte, e exigências da integração dos sistemas fontes. Mas , as atividades de conversão de dados, porém, raramente mudam. A ordem das atividades pode até mudar, mas nenhuma delas pode ser eliminada. A qualidade dos dados de armazém depende da atenção detalhada dada a cada uma das atividades de conversão de dados.

3.2.3. Estratégias de Migração de Dados – 3ª abordagem

Uma outra abordagem sugerida por [SHEP99] é a estratégia de migração de dados. Esta estratégia consiste de dois processos perfilamento e mapeamento de dados, e tem como objetivo a completa compreensão dos dados fontes desde que os processos envolvidos sejam bem aplicados. Esta estratégia para análise de dados, tem como propósito migrar dados dos sistemas legados para o *Data Warehouse*, podendo ser associada a ferramentas de migração de dados para executar os processos de extração, limpeza, transformação e carga dos dados.

3.2.3.1. Técnicas convencionais em Migração de Dados

Problemas e Armadilhas

A abordagem convencional para perfilamento de dados e mapeamento inicia com um grande grupo de pessoas (analistas de dados e de negócio, administradores de dados, administradores de banco de dados, projetistas de sistemas, etc). Estas pessoas se encontram em uma série de desenvolvimento de aplicação em comum e tentam extrair informações úteis sobre o conteúdo e estrutura das fontes de dados legados, examinando documentação antiquada, copiando livros *COBOL*, metadados inexatos e, em alguns casos, os próprios dados físicos. Perfilamento de dados legados deste modo é extremamente complexo, de intensivo tempo e propenso a erros. E quando o processo está completo, só uma compreensão limitada dos dados fonte é alcançada. Neste ponto, de acordo com o quadro de fluxo de projeto, o analista de dados passa para a fase de mapeamento. Porém, desde que o dados fonte são interpretados como pouco produtivos e as conclusões sobre isto estão em grande parte baseado em suposições em lugar de fatos, esta fase resulta tipicamente em um modelo de dados e um conjunto de

especificações de mapeamentos inexatos. Baseado nesta informação, os dados são extraídos, limpos, transformados e carregados no banco de dados designado. Não surpreendentemente, em quase todos casos, o novo sistema não trabalha corretamente a primeira vez . Desta forma o processo recomeça e tudo é feito novamente: projeto, codificação, carga e testes; incorrendo tempo significante e custo altíssimo. Na pior das hipóteses, o projeto é cancelado e a empresa prefere viver com um ineficiente mas sistema de informação parcialmente funcional em lugar de incorrer em custos contínuos de um "projeto de migração de dados infinito".

Objetivando levar projetos de migração de dados de corporações à conclusão próspera já na primeira tentativa de construção. É proposta a estratégia de migração de dados. Esta estratégia consiste em um processo de dois passos: perfilamento e mapeamento.

O processo de perfilamento de dados estuda os dados fontes completamente para entender seu conteúdo, estrutura, atributos e integridade. Constitui-se de três passos seqüênciais. Ao passo que, o processo de mapeamento de dados é um conjunto preciso de especificações, desenvolvido com base no perfilamento dos dados, e também constitui-se de três passos sequenciais.

3.2.3.2. Perfilamento dos dados

No p*erfilamento dos dados*, as fontes de dados são perfiladas em três dimensões: Colunas (perfilamento de coluna),Linhas (perfilamento de dependência) e tabelas (perfilamento de redundância).

- O *perfilamento de Coluna* permite ao analista descobrir e analisar problemas da qualidade do conteúdo dos dados e avalia discrepâncias entre os deduzidos, verdadeiros metadados e os metadados documentados. Para tal, os valores em cada coluna ou campo de dados fonte são analisados e características detalhadas são deduzidas para cada coluna, inclusive tipo de dados e tamanho, faixa de definição de valores, freqüência e distribuição de valores, cardinalidade e nularidade e características de singularidade.
- O *perfilamento de Dependência* identifica chaves primárias e que dependências não esperadas são suportadas pelos dados. Identifica também dependências de

áreas que são verdadeiras a maioria do tempo, mas não todo o tempo, e usualmente são uma indicação de um problema de qualidade de dados. O processo consiste em analisar dados por linhas comparando valores em cada coluna com valores de uma outra coluna e deduzindo todas as relações de dependência que existem entre atributos dentro de cada tabela.

- O *perfilamento de Redundância* compara dados entre tabelas de mesma ou diferente fonte de dados e determina quais colunas contêm sobreposição ou conjunto de valores idênticos. Identificando atributos que contêm a mesma informação mas com nomes diferentes (sinônimos) e atributos que têm o mesmo nome mas significado de negócio diferente (homônimos). Ajudando a determinar quais colunas são redundantes e podem ser eliminadas e as que são necessárias para conectar informação entre tabelas. Este perfilamento elimina processamento overhead e reduz a probabilidade de erro no banco de dados designado.

É importante saber que cada passo é sempre realizado sobre os dados obtidos nos passos anteriores já executados e que devido a complexidade, trabalho e recursos requeridos os processos de perfilamento de dependência e de redundância, não podem ser realizados manualmente.

Os resultados do processo de perfilamento dos dados podem ser usados para completar a segunda fase do projeto de migração – o mapeamento de dados.

3.2.3.3. Mapeamento dos dados

O *Mapeamento de dados* constitui-se de três passos: normalização, modelo expandido e mapeamento de transformação.

A *normalização* consiste na construção de um modelo baseado no relacional completamente normalizado e completamente suportado pela consolidação de todos os dados, garantindo que o modelo de dados não falhará.

O *modelo Expandido* é um processo que envolve a modificação do modelo normalizado pela adição de estruturas para apoiar novos requisitos ou adicionar índices e desnormalizar as estruturas para aumentar desempenho.

O *mapeamento de Transformação* consiste em criar um conjunto de mapas de transformação para mostrar os relacionamentos entre colunas nos arquivos fontes e tabelas no modelo expandido, inclusive fluxos de atributo-para-atributo. Idealmente, estes mapas de transformações facilitam a captura de requisitos de limpeza e de transformação e provê informação essencial para os programadores criarem rotinas de conversão para mover dados da fonte para o banco de dados designado.

Uma boa metodologia de perfilamento e mapeamento de dados levará em conta o escopo inteiro de um projeto, sendo possível verificar se os objetivos empresariais do projeto não são suportados pelos dados.

Com o estabelecimento de um perfilamento de dados e estratégia de mapeamento sólido, pode-se executar com rapidez as tarefas altamente complexas necessárias para alcançar uma compreensão completa de dados fonte, em um nível de compreensão que simplesmente não pode ser alcançada por processos convencionais e técnicas de consultas semi-automatizadas.

Estes processos quando realizados corretamente, causam uma significativa redução do risco de projeto, além de habilitar valiosos recursos para redirecionar outros projetos produtivos da empresa, proporcionando como resultado mais participação de decisões de negócios que tipicamente significam maiores lucros e rendas.

3.2.4. Tecnologias de movimento de dados - 4ª abordagem

A quarta abordagem é proposta por Hill em [HILL98] e trata da utilização de tecnologias de movimento de dados na preparação de dados para o *Data Warehouse*. Esta abordagem não diferentemente das demais irá abordar as estratégias utilizadas para preparar dados para o *Data Warehouse*, enfocando as tecnologias utilizadas.

Segundo [HILL98] : Um dos maiores desafios para a equipe do projeto de um *Data Warehouse* é a extração, consolidação e transformações de dados para povoar o *Data Warehouse*. Um desafio que requer quantidades enormes de tempo e recursos, requerendo a reconciliação de modelos de dados diferentes que executam na organização do ambiente operacional altamente heterogêneo. Para superar estes desafios a utilização de tecnologias de movimento de dados na preparação de dados para o *Data Warehouse*, mantendo qualidade de dados e minimizando a inconsistência de dados e

riscos de integridade é proposta. Tais tecnologias são máquinas de transformação de dados, ferramentas de limpeza de dados e técnicas de captura de dados modificados que suportam o potencial para reduzir estes desafios.

O primeiro passo é saber quais estratégias os usuários podem utilizar para efetivamente preparar dados para o *Data Warehouse*.

O processo de aquisição de dados é bastante complexo, contribuindo vários fatores, tais como, o fato da maioria das organizações possuirem uma média de oito diferentes SGBDs operacionais, com mais de 50 bancos de dados dos quais os dados devem ser extraídos. O que requer para acesso a estes bancos de dados heterogêneos, habilidades especiais e tecnologias para lidar com diferente sintaxe e semântica de SGBD. Além do fato de que o entendimento da modelagem dos dados operacionais e o significado de seus elementos de dados constituem um intenso esforço analítico.

3.2.4.1. Estratégias para preparar dados para o *Data Warehouse*

As estratégias utilizadas para preparar os dados para o DW são:

- *Identificação dos dados* – busca identificar a fonte operacional mais apropriada da qual elementos de dados podem ser adquiridos. Como estes elementos de dados estão normalmente contidos em várias fontes; a tarefa de identificação dos dados, envolve muita análise para entender onde existem dados, em que formato, onde há duplicação, onde existe incremento de dados valiosos e qual fonte é mais fidedigna.
- *Aquisição dos dados* - a aquisição dos dados é a coleção física dos dados adquiridos dos sistemas fontes previamente analisados, desta forma esta fase só pode ser iniciada quando a análise que ocorre na identificação dos dados está completa.
- *Limpeza dos dados* – os clientes normalmente limpam os dados antes de carregar, o que resulta em menos erros na carga corrente.
- *Transformação dos dados* – Nesta fase os dados são transformados em conteúdo descritivo do negócio para serem carregados.
- *Atualização do DW* – Após o povoamento inicial do *warehouse*, uma estratégia de atualização do *Data Warehouse* também deverá ser desenvolvida.

3.2.4.2. Complexidades dos processos de migração de dados

Estabelecendo que toda a complexidade dos processos de extração, transformação e integração dos dados, dependem do número e variedade de fontes de dados. O produto matemático do número de fontes de cada tipo é proposto como uma aproximação de complexidade de primeira ordem. Se isto é aproximadamente preciso, quanto mais tipos de fonte de dados introduzidas, mais complexa a implementação do resultado do processo de aquisição de dados se torna. Para integrar dados de aplicações de legado com aplicação cliente/servidor desenvolvida recentemente, os usuários acrescentam um perigoso nível de complexidade que produzirá à extração e integração, tarefa quase impossível para executar métodos de aplicação tradicionais sem uma grande quantidade de recursos e um complexo processo de multipassos que serão difíceis manter. Para enfrentar tais desafios empresas têm que ou aumentar ferramentas que geram código (por exemplo, *ETI*) com aplicações desenvolvidas personalizadas ou utilizar tecnologia nascente *TE* (por exemplo, *Constellar*), fazendo com que continuem gastando recursos adicionais porque o problema não pode ser completamente resolvido com tecnologia obsoleta. Logo, habilidades para SGBD pré-relacional devem ser mantidas se os dados forem requeridos para extração, transformação e carga de dados no *Data Warehouse*.

3.2.4.3. Atualização do *Data Warehouse*

Quando atualizando o armazém de dados, considere com que freqüência e também com que granularidade (i.e., atualiza, insere, deleta, adiciona) é feita a carga incremental. A tendência é que as atualizações sejam noturnas. Para facilitar atualizações se deve criar uma área de estágio relacional. Uma área de estágio de dados (ou área de organização de dados) provê muitos benefícios, tais como:

1. Tornar homogênea as diferenças entre os bancos de dados operacionais discrepantes;
2. *Buffers* comparando as diferenças em disponibilidade das fontes;
3. *SQL* sendo usada para comparar relacionalmente, combinar e purgar registros em lugar de chaves comuns.

Superconjuntos de Tecnologia de Movimento de Dados

As tecnologias de movimento de dados variam da mais simples (programas de transferência de arquivo) para a mais completa ferramenta (transformação de dados). Formas mais avançadas são na verdade superconjuntos das formas mais simples. Estas tecnologias posicionam-se de acordo com duas dimensões importantes: transformação e escalabilidade. Escalabilidade se refere à habilidade da tecnologia para dirigir múltiplas fontes e destinos diversos. Os produtos menos funcionais somente manipulam transformações da forma uma- para- uma, considerando que mais produtos escaláveis podem manipular transformações muitas- para- muitas. Organizações enfrentam múltiplas e diversas exigências de movimento de dados para projetos diferentes, ou esses com múltiplas fontes e destinos discrepantes, deveriam considerar as ferramentas de transformação mais funcionais que provêem mais rica programação e flexibilidade de execução para conhecer um conjunto amplo de exigências. Projetos de *Data Warehouse* caracteristicamente têm múltiplas fontes operacionais com transformação complexa requeridas para criar dados descritivos empresariais. Porém, ferramentas de transformação hoje geralmente executam um processo *batchlike*, manipulando um *batch* de inserções. A habilidade destas ferramentas para executar captura de dados modificados é tipicamente muito limitada para somente uma fonte relacional. Então, organizações continuam procurando por um método apropriado para atualizar o *Data Warehouse* em uma base incremental (incluindo atualizações, deleções e inserções).

3.2.4.4. Alternativas de Movimento de dados

As tecnologias de movimento de dados alternativas são posicionadas ao longo de duas dimensões: complexidade e dados correntes. Complexidade se refere ao número de fontes, o grau de heterogeneidade e o grau de transformação. Quanto mais dados são transformados, maior a exposição à consistência e integridade de dados. As alternativas de movimento de dados são:

- *Propagadores de dados* propagam dados entre SGBDs heterogêneos ou esquemas.
- *Replicação* que ocorre entre SGBDs homogêneos e esquemas.
- Ferramentas de *transformação* de dados que mudam os próprios valores dos dados e até mesmo o significado dos dados (semântica). Aplicando algoritmos lógicos, usando uma linguagem de programação diferente de uma *DML* (i.e., *SQL*) para transformar dados para uso em outro ambiente de aplicação.
- *Sincronização de dados* é um perfil de aplicação, não uma tecnologia. A verdadeira sincronização requer protocolo *commit* de duas fases. A replicação reconcilia atualizações para dois ou mais bancos de dados, embora momentaneamente, pelo fato de mudanças continuarem acontecendo nos bancos de dados. A replicação e consulta de mensagem podem ser usadas para sincronização específica de aplicação. Embora tecnologia e regras de negócio devam ser programadas para dirigir a reconciliação, e ser decisivas ao sucesso da aplicação.
    - *Gerenciamento de cópia* é a mesma tecnologia da replicação. A diferença está na implementação, a qual é normalmente uma coleção de *schedules* via *SQL* de um consistente conjunto de dados. É útil quando a ordem de transações não é critica e pequeno overhead no banco de dados de fonte é preferido.

3.2.4.5. Replicação na arquitetura de *Data Warehouse*

A replicação (duplicação) de banco de dados pode ter um papel na arquitetura de DW. Quando o esquema de um *Data Mart* local usa um subconjunto do DW (talvez

com outras tabelas desnormalizadas), podem ser usados replicações para atualizar o *Data Mart*. Esta abordagem assume que os dados operacionais já foram adquiridos, limpos e transformados. Um segundo cenário é quando alguns dados locais, tais como dados de vendas locais, precisam ser consolidados por cruzamento geográficos. Em organizações altamente distribuídas, são vistos DW incorporados que só contém algumas áreas de assunto - esses verdadeiramente separados pelo empreendimento como contabilidade e dados financeiros – com DW geograficamente locais para partes diferentes do negócio do empreendimento. Neste cenário, dados locais podem precisar ser consolidados e requerer conversões correntes. Organizações aplicam a replicação neste cenário, avaliando cuidadosamente se podem ser implementadas regras de triangulação.

Tendências que afetam preparação dos dados para o DW :

- Todos os anos mais dados são movidos para relacional;
- Cada vez mais dados são administrados através de pacotes de aplicações;
- Vendedores de transformação de dados acessam fontes relacionais, ou perguntam por dados para ser "*flattened*".
- Vendedores de extração de dados são pressionados e não podem prontamente simplificar o problema de acesso aos dados legados.
- Janelas em batch estão desaparecendo, volumes em *OLTP* estão aumentando;
- Mais que 80 por cento dos usuários de DW codificam a sua própria rotina de extração.

A tarefa de acessar dados pré-relacional permanece urgente e inflexível como sempre. E inovações não têm acontecido para fazer esta tarefa mais fácil. Os vendedores de ferramentas de extração tradicionais estão em grande maioria longe desta tarefa. Considerando o número de DW em produção, a adoção de ferramentas de terceiros tem sido muito baixa. (Embora a adoção de ferramentas para povoar DW esteja acelerando).

Acessar bancos de dados relacionais é muito mais atrativo. Mesmo quando o banco de dados é unicamente transacional, existe pelo menos um catálogo. E quando este catálogo não é muito fidedigno como um indicador de estrutura de dados, o vendedor deve fabricar evoluções para simplificar acesso aos dados.

3.2.4.6. Arquitetura do mecanismo de transformação

Vendedores de produtos de extração de dados (por exemplo, *ETI*) fornecem ferramentas para simplificar a arquitetura de povoamento de um DW. Ferramentas geradoras de código *ETT* provêem aos usuários com um ambiente *CASE* para gerar aplicações batch orientadas a *COBOL*, executando a extração e transformação de dados de sistemas operacionais para ser incluído no *Data Warehouse*. Para algumas fontes e exigências de transformação limitadas, ferramentas tais como estas executam adequadamente e dão um retorno no investimento feito pela empresa. Porém, como o número de fontes cresce (i.e., maior que cinco) e a complexidade das transformações aumenta, estas ferramentas tiveram dificuldade em demonstrar a habilidade para escalar. Máquinas de transformação provêem uma habilidade melhorada para manipular um número grande de fontes de dados e complexidade em transformações. A força destas ferramentas é a capacidade para definir e dividir os processos (por exemplo, integração e transformação de dados) em tarefas separadas que usam um cubo de dados como um armazenamento transitório, ou área de estágio (organização) de dados operacionais. Porém, todas as ferramentas baseadas em máquinas de transformação são limitadas naturalmente a acessar SGBDRs, flat files (arquivos simples), ou bancos de dados acessíveis *ODBC – Open Database connectivity* - (i.e., planilhas eletrônicas *Excel*); outras fontes devem ser descarregadas manualmente ou devem ter acesso via um *gateway*. Desta forma uma área de estágio de dados deve ser parte do processo de *ETT*, sendo desenvolvido manualmente ou com ferramenta.

3.2.4.7. Tecnologias para atualização de *Data Warehouse*

Uma tendência que tem acelerado nos últimos anos é a construção de ODS. Esta estrutura de apoio á decisão é usada tanto para investigação, como para propósitos de atualização. Embora freqüentemente construído para suportar as exigências da organização na administração de relação de clientes, o *ODS* também serve como uma única fonte de rede de mudanças que aconteceram nos sistemas operacionais desde que o DW sofreu a última atualização. O *ODS* é um modelo de dados único que reconcilia as diferenças semânticas inerente nos sistemas de *OLTP* - *stovepiped*. Normalmente,

um corretor de mensagem é usado para integrar as aplicações legadas *OLTP* com o *ODS*. Corretores de mensagens podem capturar os eventos do negócio ou transações no ambiente da aplicação de *OLTP* à camada de aplicação (usando uma integração de níveis *API*) e propagar os eventos, com ou sem transformação semântica como exigido, para o *ODS* receptor. Este estilo de integração pode suportar eventos batch ou eventos singulares, dependendo das exigências para informação oportuna. Um *ODS* baseado em SGBDR suporta o uso de tecnologia de replicação de banco de dados para capturar um conjunto consistente de dados modificados que atualizam o *Data Warehouse*. Esta abordagem é recomendada em cima de atualizações individuais capturadas em cada uma das muitas fontes *OLTP* e tentando as reconciliar diretamente no *Data Warehouse*. O *ODS* é essencialmente uma área de estágio para o DW.

Em todas as abordagens foram citados mesmo que superficialmente a preocupação em garantir qualidade de dados durante o processo de migração de dados. A seção a seguir trata deste tema, procurando descrever como é possível buscar a qualidade de dados para o *Data Warehouse*.

3.2.5. Qualidade de Dados

No ambiente empresarial competitivo de hoje, novos desafios para aquisição de clientes ocorrem diariamente. Para se destacar, companhias tem utilizado soluções de apoio à decisão, tais como, *Data Warehouse* ou *Data Mart*, a fim de ter uma sólida fundamentação de informação, na qual possa tomar suas decisões de negócios.

Um dos maiores obstáculos que bloqueiam o sucesso de muitos projetos de *Data Warehousing* é a precisão dos dados. Isto porque de 10 a 20 por cento dos dados usados estão de certa forma incompletos e corrompidos. È prática comum os registros em um banco de dados conter algum tipo de informação que necessita ser corrigida [IDCE99].

Embora limpeza de dados possa levar a muitas formas, a maioria dos exemplos importantes de limpeza de dados da lista prévia de aplicações surge da necessidade por boas descrições de coisas tangíveis como clientes, produtos, procedimentos e diagnoses. O mercado atual e a tecnologia atual para limpar dados são pesadamente enfocados em listas de clientes. E uma das áreas mais cruciais de qualidade de dados é a informação

de cliente. Onde a precisão da informação é obtida com a incorporação de qualidade de dados em cada passo de extração, transformação,consolidação, e manutenção dos dados.

A qualidade de dados é especialmente importante para a consolidação precisa, porque permite reconhecer e entender relações de cliente.

Para poder trabalhar com qualidade de dados é necessário que seja determinado quais dados são importantes à empresa. Para então definir a técnica de qualidade de dados a ser aplicada.

Em se tratando de qualidade de dados e consolidação de produtos, normalmente se encaixam uma das duas categorias: soluções referenciadas a dados que combinam tabelas de referência com algoritmos sofisticados, e soluções não referenciadas a dados que utilizam somente algoritmos. Em *Data Warehouse* onde dados de clientes são essenciais, a utilização de software referenciado a dados é mais efetiva. Este tipo de software caracteriza uma base de conhecimento extensa de dados empíricos que permite aumentar e melhorar a qualidade da informação.

A qualidade de dados é alcançada em três fases: limpeza, comparação, e consolidação. Na fase de limpeza de dados, o dado é analisado gramaticalmente, corrigido, unificado, e melhorado para comparação precisa. Na fase de comparação, comparações são feitas internamente e cruzando os dados fontes para localizar informação semelhante. Os dados comparados são consolidados e colocados em um *Data Warehouse*, *Data Mart*, ou outra área de armazenamento de dados.

3.2.5.1. Análise gramatical

O primeiro componente crítico em limpeza de dados é analisar gramaticalmente os dados. O propósito deste processo é tornar mais simples a correção, padronização e comparação dos dados. Isto porque permite localizar, identificar,e isolar elementos individuais de dados no arquivo de clientes. Proporcionando comparação entre componentes individuais ao invés de em muitas *strings* de dados.

Analisar gramaticalmente é um passo vital para a fase de limpeza e comparação e existem vários obstáculos para analisar gramaticalmente e isso pode dificultar depois uma comparação de sucesso. Os problemas mais intensos são as discrepâncias encontradas nos metadados. Por exemplo, a informação em um campo pode não

corresponder ao seu perfil de metadados. Não permitindo desta forma determinar se campos de fontes de dados múltiplas possuem as mesmas características.

Outros obstáculos incluem:

- dados colocados em campos errados,por exemplo, dados de endereço, no campo nome ;
- dados flutuantes, que são aqueles que podem estar contidos em diferentes campos de registros;
- informação estranha: os dados podem conter campos irrelevantes ou em branco;
- palavras atípicas: registros podem incluir étnico, multicultural, e nomes hifenizados, títulos incomuns, nome de negócio abreviado e acrônimos específico de industria;
- estruturas incompatíveis.

Mesmo estando os dados analisados gramaticalmente, é necessário saber se são dados precisos. Logo uma outra fase do processo de limpeza de dados deve ser feita – a correção.

3.2.5.2. Correção dos dados

Quando os dados são oriundos de várias fontes, pode-se encontrar:

- Formatos de campos incompatíveis, se existe um obstáculo para não analisar dados, combinar todas as fontes de dados em um *Data Warehouse* é tarefa muito difícil;
- Variações em abreviações, formatos, etc., por causa de preferências individuais da pessoa que dar entrada na informação;
- Grafias erradas causadas por semelhanças fonéticas durante entrada de dados por telefone.
- Informação Antiquada devido a mudanças de nome e de endereço;
- Transposições que são o resultado de erros teclados.

O único modo para corrigir e verificar dados prosperamente é usar software que referencia uma fonte de dados secundária fidedigna. Em muitas instâncias, correção é

usada somente para preparar dados por comparação, deixando os registros originais inalterados.

3.2.5.3. Padronização

A padronização, é o processo em limpeza de dados, que permite organizar informação de cliente em um formato preferido e consistente. Alguns dos maiores desafios para padronização precisa de dados de cliente, incluem:

• abreviações Incompatíveis– por exemplo:

Internacional Ceifeira,

Intl. Ceifeira,

Interntl. Ceifeira,

Internatl. Ceifeira

• grafia errada e variante de ortografia, isto é, *Kwik, Quik, Quick.*

Quando limpando certos tipos de dados (nomes, nomes empresariais, títulos profissionais, etc.), padrões de comparação facilitará comparação mais próspera.

Padrões de representações de comparação típicas de um elemento de dados, só pode ser nomeado por software de padronização sofisticado.

Alguns softwares também podem padronizar outras informações de clientes, tais como: pré-nomes, nome divulgado, títulos, e localizações empresariais. Por exemplo, de "Doutor para Dr.", "Júnior" para "Jr.", "*Floor*" para "Flr.", etc. também pode identificar gêneros, baseado em dados de nome empíricos.

Unificando os elementos de software de padronização Sofisticado pode-se indicar padrões de comparação próspera para elementos tais como: nomes pessoais e de negócio. Por exemplo, padrões de comparação:

General Electric GE

Gen. Electric GE

Al : Albert, Alfred, Alan, Alphonse, Almon, Alexander.

3.2.5.4. Aperfeiçoamento dos dados

O aperfeiçoamento dos dados é o passo final em limpeza de dados, consiste em juntar dados novos e completar informação que falta. O tipo de informação adicionada pode incluir:

• dados Demográficos – por exemplo, idade, presença de crianças,renda, nível de educação, e volume de vendas, número de empregados, e código SIC para negócios.

• dados Geográficos – por exemplo apartamento ou números de suites, elementos de endereço que falta, número de telefone, código de município,e distritos políticos

• dados comportamentais – por exemplo, mérito de crédito, meio de comunicação preferido.

• dados psicográficos – por exemplo, passatempos, interesses, e afiliação política

• Evento de dados direcionados – por exemplo, matrimônio, nascimento de uma criança.

• dados computadorizados – por exemplo, avaliações de crédito.

O aperfeiçoamento pode ser realizado enviando os dados de cliente para outra firma para processar, comprando uma fonte externa de dados de cliente, ou inspecionando os clientes e atualizando a informação deles/delas manualmente. Porém, estas alternativas podem apresentar alguns desafios adicionais, tais como:

• Processo Externo pode ter tempo e custo-proibitivo.

• Os registros de cliente podem conter dados muito sensíveis para enviar a uma firma externa para processar.

• recursos Internos e perícias podem ser insuficientes para integrar uma fonte de dados externa.

• Clientes podem ser difíceis para alcançar ou pouco dispostos para prover informação adicional.

Quando a informação é limpa e padrões de comparação são utilizados, as representações duplicadas podem ser eliminadas e toda a informação sobre cada cliente individual ou uma família inteira pode ser consolidada. Um dos maiores desafios em

comparação é criar um sistema que incorpore as regras de negócios das empresas. Alguns desafios adicionais para comparar dados de negócio-para-negócio incluem:

• Fusões de Companhias, aquisições, ou mudanças de nome incorporadas;

• Relações entre divisões subsidiárias, e corporações de origem;

• acrônimos Empresariais (por exemplo, *NASDAQ* ou *NYNEX*) ;

• Inicialismo – a primeira letra de uma ou mais palavras em um título ou frase que são soadas um por um (por exemplo, *AT&T* ou *CIA*).

Uma vez os registros comparados, e identificadas relações entre clientes, uma visão consolidada pode ser feita.

Há dois métodos para consolidação: o primeiro processo de consolidação combina todos os dados em qualquer determinado cliente usando todas as fontes de dados disponíveis. O segundo processo revela vínculos os clientes da empresa – é identificação de clientes.

Em qualquer caso, deve ser escolhida uma solução de qualidade de dados que inclua:

• Opções de padronização de clientes;

• A habilidade para reter dados originais e economizar informação corrigida;

• opções de saída Flexíveis.

O software de qualidade de dados mais efetivo também pode analisar gramaticalmente nomes, títulos, localizações empresariais, nomes empresariais, e condições financeiras como fiduciário, aposentado, etc. Usando dados empíricos e tabelas de modificação de usuário estes sistemas podem localizar com mais precisão dados flutuante, fora de campo, ou incorretos analisados gramaticalmente.

Quando escolhendo sistemas de limpeza de dados referenciados a dados, deve-se verificar a habilidade para:

• Reconhecer nomes de rua formal e informal, endereços múltiplos (por exemplo, caixa de correio e endereço de rua no mesmo registro),e nomes de cidade.

• Minimize a aprendizagem de usuários usando uma significante base de dados pré-definida.

• Realizar aperfeiçoamento com dados já integrados nos arquivos de referência.

• *Flag* antiquado ou dados de endereço inválidos (por exemplo, rural - endereço de rota rural convertido para endereço ‘ 9-1-1 ' , ou um endereço não deliberado).

• Acessar e modificar tabelas de referência

• Determinar padrões de comparação.

Há uma variedade de maneiras para disponibilizar sistemas de comparação, cada qual oferecendo um modo diferente para chegar a uma correspondência:

• Comparações de código chave - executa comparações idênticas usando alguns primeiros caracteres em um ou mais campos. Este método primitivo raramente é praticado porque usa só um pequeno subconjunto dos dados que podem resultar em muitas falsas comparações.

• Sondagens - descobrem semelhanças fonéticas, tais como, ‘f ' e ‘Ph’. Por exemplo: de *Quick* e *Kwik*. Estes erros resultam freqüentemente de dados recebidos através de telefone, particularmente com dados, não podendo ser unificado. Porém, Sondagem é inadequado como uma solução exclusiva porque só pode descobrir erros fonéticos.

• Semelhança de comparação – também chamado comparação *fuzzy* – pode identificar comparações computando um grau de semelhança entre dois componentes discretos. Porque comparações idênticas não são requeridas, isto pode ajustar para fonético, tipográfico, e erros de transposição. A semelhança de comparação é considerada o melhor amplamente método de comparação. Isto é especialmente valioso para dados que não podem ser padronizados, tais como últimos nomes, nome de negócios e números de casa.

• Comparação ponderada - pode ser usada junto com a comparação de sondagem ou semelhança. Permite indicar a importância relativa de campos que determina uma comparação.

Comparação de semelhança considera todos caracteres em um campo e a posição deles para determinar o grau de uma comparação. Por exemplo, o grau da correspondência, neste caso em particular, é indicado pela pontuação de semelhança. Por exemplo:

No endereço 1001 St. de Rosa:

1001 e 101 têm uma pontuação de semelhança de 85%

1001 e 1010 têm uma pontuação de semelhança de 75%

1001 e 1025 têm uma pontuação de semelhança de 50%

Algoritmos de propósito especiais que são extensões de comparação de semelhança aplicam a lógica de exceção (regras se/então) para regras de correspondências tradicionais. Há quatro categorias de algoritmos de propósito especiais:

• Caso especial de campo de lógica personalizado - compara técnicas para campos específicos. Por exemplo, estes algoritmos são usados para identificar comparações entre acrônimos ou inicializações e os nomes empresariais completos deles, ou componentes numéricos dentro de nomes.

• Caso geral de campo lógica aplicada a lógica de comparação adicional - quando encontra certas anomalias, tal como campos com espaço em branco. Em Comparação de componentes discretos, permite especificar quando campos em branco deveriam ser considerados correspondentes a campos que contêm dados.

• Caso especial de múltiplo campo de lógica - ajusta ponderações que depende dos dados achados em conjuntos específicos de campos. Como por exemplo, *householding* este método nomearia um valor mais alto para o campo de nome, quando o endereço é um apartamento complexo e o número da unidade é um campo em branco.

• Caso geral múltiplo campo de lógica - executa uma segunda comparação quando encontra anomalias específicas, indiferentemente dos campos nos quais eles são achados.

Uma abordagem combinada de semelhança incorporada, ponderação, e algoritmos propostos especiais – é normalmente melhor.

3.2.5.5. Sistemas de consolidação

Os processo de limpeza de dados e comparação conduzem a um resultado final: consolidação de dados precisa. Para construir isto, será preciso uma solução de consolidação flexível para combinar existência de dados operacionais e manutenção de alimentação de entrada de dados. Alguns componentes de soluções chaves para consolidação evidenciadas permitem:

• Priorizar fontes de dados entrantes - Bancos de dados próprios são normalmente mais fidedigno que dados comprados ou alugados porque eles apresentam-se mais atuais. Com lógica de campo de caso especial, uma comparação entre um nome completo e suas inicializações pode ser identificada ou entre números em um nome e a ortografia completa deles.

• Priorizar campos – campos que foram limpos e verificados e verificados tendem a ser mais fidedigno que aqueles que não passaram por este processo.

• Manter fontes de dados originais. Metadados completos permitem localizar erros de dados ou discrepâncias na fonte.

• Identificar dados incertos ou perdidos - Uma vez identificado, se possível pedir informação diretamente do cliente ou de uma outra fonte válida.

3.2.5.6. Considerações finais

A Qualidade de dados é essencial para comparação e consolidação de qualidade. É crítico determinar quais dados são mais importantes, e escolher a ferramenta mais adequada. Mas, quando trabalhando com dados de cliente, uma solução referenciada a dados é melhor.

Com uma variedade de soluções de comparação disponível, deve-se procurar uma que combine tipos diferentes de algoritmos de comparação.

O melhor curso de ação é escolher ferramentas que oferecem a flexibilidade e precisão do projeto de demandas. Implementando estas ferramentas, é possível afiançar uma fundação sólida para construir relações de cliente uma por uma.

3.2.6. Ferramentas

Para se ter sucesso no processo de migração de dados, bem como, se obter qualidade de dados é necessário utilização de ferramentas. Esta seção versa sobre ferramentas *ETT*, aspectos de avaliação de ferramentas e algumas ferramentas disponíveis no mercado que atendem a estes aspectos.

Vendedores de ferramentas geradoras de códigos *ETT* (*Carleton, ETI, Platinum Tecnology e Prism Solutions*) enfatizam funcionalidades de extração, considerando que

os vendedores de máquina de transformação (por exemplo, *Constellar, Informatica, Ardent*) enfatizam funcionalidades de transformação. Comparando as forças destas duas categorias de ferramentas, se verifica que são mais complementares que competitivas.[HILL98]

3.2.6.1. Critérios para avaliar produtos de migração de dados.

Os seguintes critérios de avaliação podem ser usados quando avaliando ferramentas de extração e transformação como uma base para construir específica, detalhada lista de característica.

Funcionalidade de extração é importante para empreendimentos com uma ou mais das seguintes características:

1) Projetos e requisitos de transformação de dados múltipla;
2) mais que três fontes de dados pré-relacionais heterogêneas com grandes volumes;
3) moderado para registro de alta comparação, combinando necessidades de integração.

Outras organizações podem anteceder natural funcionalidade de extração e enfocar em funcionalidade de transformação contanto que a ferramenta escolhida proveja administração de metadados. Administração de Metadados tem o potencial para reduzir dramaticamente as especificações necessárias por desenvolver novos processos. Em casos onde ou as fontes são bastante homogêneas (por exemplo, principalmente SGBDR - dados gerenciados) ou a organização está disposta para processos *handcraft* para descarregar as fontes, a velocidade de aprendizagem e desenvolvimento mais rápido de processos de transformação farão para máquinas de transformação uma aproximação viável.

Quatro estilos de Integração de Aplicação

Quatro estilos de projeto de integração de aplicação de *post hoc*. As escolhas são binárias; a integração é realizada ou nos níveis de dados da aplicação (i.e., um arquivo ou o banco de dados) ou no nível de evento (i.e., normalmente uma transação ou mensagem criada pela própria lógica de aplicação). Ao longo do eixo vertical,

integração é realizada individualmente nos dados ou base de evento ou é realizada em um conjunto ou coleção de elementos de dados ou eventos. Um método de integração baseado em um conjunto é apropriado quando um grau mais alto de latência pode ser tolerado entre a aplicação enviada e recebida. Os estilos individuais são mais apropriados quando mais baixa latência é desejável. Uma abordagem de integração baseada em um conjunto é mais apropriada para a maioria das formas de aplicações de SAD (*reporting*, DW, DM) e para ambientes de SGBD distribuídos, porque as aplicações tipicamente possuem características tais como: a) atualização em batch, b) médio para alta latência é aceitável, e c) transformação é feito para um conjunto de elementos relacionados (i.e., todos os clientes). Porém, uma tendência que se desenvolve é relativa a uma "estratégia de latência zero," onde a meta é consciência instantânea e resposta apropriada para eventos por um empreendimento inteiro ou além. Em nossa visão, uma estratégia de "latência zero" é qualquer estratégia que explora a troca imediata de informação cruzadas por limites técnico e organizacional para alcançar benefício empresarial. Desta forma, organizações deveriam selecionar uma tecnologia para integração em batch e uma integração em real time.

3.2.6.2. Critérios para avaliar ferramentas de transformação de dados

Quando escolhendo ferramentas de transformação de dados devem ser adotados alguns critérios, tais como para avaliar a ferramenta [MORI98]:

- Suporte de transformação de dados - uma das primeiras coisas para determinar sobre suporte de transformação de dados é a habilidade da ferramenta para definir, ler, digitalizar, e extrair dados de sua aplicação fonte.
- Áreas de compatibilidade primárias - a linguagem de programação que a ferramenta usa e suportes (como *Cobol*, *C*, ou *C++*),as linguagens de definição de dados (como *DB2*, *Oracle*, *IMS*, ou *IDMS*).
- Habilidade da ferramenta para importar estrutura de dados de um catálogo de *DBMS* ou uma *CASE* ou produto de repositório. Verificar se quando a ferramenta não pode ler em todos suas estruturas de dados, ela dá a opção de entrar nas definições de estrutura de dados manualmente.

- Quais tipos de transformação a ferramenta pode suportar e o método que utiliza para este suporte.
- Decidir adequadamente o suporte a conversões de tipos de dados que a tecnologia designada requer.

Algumas das transformações exigidas mais comuns que a ferramenta deve atender são:

- Movimentos diretos (mova campo de fonte A para campo destino B);

- Campos temporários (campos calculados organizados durante o processamento, como contadores ou saldo correntes);

- Cálculos (conversão aritmética de dados, usando tanto dados designados como temporário, por exemplo, Objetivo = (2> (FonteB) / (FonteC-TempE)));

- Cálculos estatísticos (TargeA=AVE(B), onde o AVE é a função comum e B é o domínio dentro do campo de fonte para o qual a função comum é executada) ;

- Cálculos sumários ;

- Tabelas *Lookups* (Targe~=Name(B), onde Nome é o índice para uma tabela externa ou arquivo e B é o nome do campo dentro daquele arquivo) ;

- Transformações condicionais (transformações baseadas em lógica de *Boolean*, por exemplo,: Se A = X, então B, senão B = 0) ;

- Concatenação (número de telefone = Código de Área + Prefixo + Número de Linha);

- Operações de *Substring* (por exemplo, Código de Área = número de telefone (1,3));

- Conversões de tipos de dados.

- Quando a ferramenta não dirigir todas as conversões de tipos de dados, se deve verificar as capacidades que isto provê para apoiar saídas de usuários, bem como se deve verificar janelas onde pode ser encerrado e onde os usuários podem escrever código para apoiar transformações não padronizadas, programando na linguagem que a organização tem suporte.
- Deve ser analisado como a ferramenta deixa entrar e mantém as regras de transformação. A aplicação suporta mapeamento gráfico de dados? Usuários poderão executar entrada de dados apontando e clicando? A habilidade da ferramenta para prover assistência de mapeamento e atalhos tais como elementos de dados de *autolinking* com nomes comuns ou já procurando um nome dentro de dados entrados deve ser observada. Que outra ajuda de mapeamento provê a ferramenta?
- Verificar a habilidade da ferramenta para capturar lógica de seleção para processar um subconjunto dos dados fonte. Pode ser utilizada lógica Booleana para definir este subconjunto? A ferramenta suporta lógica de ponto de ruptura para processar, e quais cálculos executam dentro destes pontos de ruptura (por exemplo, cálculo estatístico e relatórios)?
- Se a transformação estará mapeando de fontes múltiplas, integração de dados e funcionalidade de sincronização é crítica. Logo se deve estar certo que a ferramenta tem a habilidade para processar arquivos de fonte múltiplos concorrentemente e registros de partida que representam a mesma instância empresarial por esses arquivos (por exemplo, quando dados para o mesmo cliente existe em arquivos de fonte múltiplos). a ferramenta pode construir um único registro ou pode fixar de registros relacionados dos registros que foram comparados por arquivos de fonte múltiplos? Se um erro é descoberto no processo de sincronização, tais como um registro perdido em um ou mais arquivos fonte , a ferramenta pode descobrir esta condição? Nesse caso, que opções dá para os usuários para identificar e corrigir esta condição (parar de processar, erro de relatório, continua processando)?
- Identifique os níveis de sumarização disponível dentro da ferramenta. Estas características são embutidas, ou os usuários as têm que desenvolver? A

ferramenta provê funções de resumo suficientes (como média, significado, e total) e funções estatísticas (como divergência padrão) para suas necessidades de análise de dados?

- Informação pode ser tirada do elemento de dados, coluna objetivo, e nível de arquivo fonte? Para cada regra de transformação, a ferramenta pode listar campo de fonte, coluna designada, e processamento de especificação? Quando um erro é descoberto ou uma condição de não mapeada é encontrada, a ferramenta pode identificar a condição claramente para o usuário para avaliação e possível correção?
- Se relatório é provido, onde no processo faz este ocorrer como a condição é descoberta ou ao término de processamento? Usuários podem usar relatórios de dados entrados na ferramenta como também dados calculados ao longo do processo de transformação (como campos temporários, cálculos, e totais de sumarização)? Novamente, se a ferramenta não suporta o relatório exigido, permitirá para os usuários sair e definir os seus próprios relatórios? Nesse caso, confira a linguagem de programação requerida para esta definição de relatório.
- A ferramenta de relatório pode fazer a ordem de execução de regras de transformação relacionadas (como todas as regras de transformação que são incluídas dentro de uma unidade de programa/compilador)? Qual ambiente de *run-time* suporta a ferramenta, e a ferramenta requer que seu repositório de metadados esteja disponível no ambiente de execução? A ferramenta gera código para as regras de transformação?

3.2.6.3. Abordagens de ferramentas de migração

Existem duas categorias de produtos de migração:

- Ferramenta produto de primeira geração que geram programas em uma linguagem de programação, como *Cobol* ou *C*.
- Ferramenta produto baseada em regras de segunda geração que usam as regras de transformação codificadas no repositório de metadados no ambiente operacional ativamente. O valor desta categoria de ferramentas é que permitem mudar facilmente as regras de transformação usadas. Porém, se deve considerar

as ramificações de usar tais ferramentas cuidadosamente, pois podem requerer um componente de *run-time* que tem acesso para o repositório de metadados.

3.2.6.4. Algumas ferramentas de migração de dados

Algumas Ferramentas existentes no mercado para efetuar a migração de dados de aplicações fontes para banco de dados destino:

DataMirror Corporation
http://www.datamirror.com /

Plataforma: *DataMirror Transformation Server* é instalado em *IBM AS/400* e também em um servidor de *NT* executando em *Microsoft SQL Server*- baseado em *Data Warehouse*.

Funcionalidade do produto: O servidor de transformação *DataMirror* provê a flexibilidade para mover facilmente e continuamente sincronizar dados selecionados entre *IBM AS/400* e *Microsoft SQL Server database* - sem necessidade para reestruturar ou reprogramar as aplicações que dirigem o negócio.

Vantagens: Permite prover acesso melhor para dados mais significantes mais rápido que antes. O servidor de transformação permite completar atividades de replicação de dados em um horário de tempo flexível. Não somente se pode reproduzir dados em tempo real, mas também ter a flexibilidade para iniciar replicação de dados em um intervalo específico (isto é, sempre 24 horas).

Desvantagens: Em um ambiente de plataformas múltiplas que une *AS/400* e *NT*, uma interface específica de plataforma é usada. Isto pode fazer a configuração e operação da replicação um pouco confusa.

Tecnologia *Acta, Inc*.

http://www.acta.com /

Plataforma: O ambiente de *Maxtor* consiste em multiprocessador *HP T570s* e *HP T520s*. *Maxtor* executa em *Oracle 7.3* e usa os Objetos de negócio com ferramentas *OLAP Essbase*. *Acta's Sales Analysis RapidMart* para *SAP* e *ActaWorks* executam em *Microsoft Windows NT* e *HP-UX* e trabalha com *SAP R/3* versão 3.0 ou mais alto.

Funcionalidade de produto: O *Rapid Mart* é um pré-pacote de *Data Mart* construido com *ActaWorks*, uma ferramenta de extração, transformação e carga projetada especialmente para construir de aplicações *ERP*. *Acta's sales analysis rapid mart* provê uma solução de armazém "fora-do-caixa" que inclui um esquema designado, mapeamento de fonte-para-designado e transformações que manipula captura de dados modificados, extração de hierarquia, recuperação de erro e outros complexos processos de *Data Warehouse*. Cada *Rapid Mart* contém *jobs* de extração de dados pré-definidos que automaticamente povoam o *Data Warehouse* com a *company's SAP R/3 data*. A análise de vendas *Rapid Mart* é construída com *ActaWorksand* pode ser facilmente encomendada para satisfazer as necessidades únicas a cada companhia.

Vantagens: *Acta's Sales Analysis RapidMart* grandemente reduziu o tempo de desenvolvimento, risco de projeto e despesa de construir um *SAP R/3* baseado em *Data Warehouse*. Cada *RapidMart* tem *jobs* pré-definidos de extração de dados que automaticamente povoam o armazém de dados com *SAP R/3* dados. Além disso, os *Rapid Mart* são encomendados e aperfeiçoados podendo ser criados rapidamente e facilmente.

Desvantagens: *Maxtor* tem requisitado características adicionais para facilitar o desenvolvimento da análise de vendas do *Rapid Mart* em produção, bem como, gerado encarecimento de *front-end*, o qual *Acta* apresentou prosperamente para versão 2.0.

Ardent Software Inc., Westboro, Mass

http: //www.ardent.com/

Plataformas: O *Ardent DataStage* 3.6, executa *Servidor*: *Solaris, HP_UX, AIX, Tru64, Windows NT*. Plataforma de Cliente*: windows 95/98, windows NT* .

Funcionalidade de produto: *DataStage* 3.6 é uma solução de extração de dados, transformação, e carga que é utilizada para estabelecer *Data Warehouse* ou *Data Mart*. Tendo tempo de implementação reduzido. *DataStage* 3.6 tem três componentes de servidores centrais e quatro ferramentas para clientes. No servidor de *DataStage* encontra-se o repositório que contém uma facilidade de armazenamento centralizada para *Data Warehouse* ou informação de *Data Mart*. *DataStage* tem capacidade para extrair dados de uma larga ordem de fontes, inclusive planejamento de aplicações de recursos de empreendimento e sistemas legados. Estas capacidades fazem *DataStage* ideal em cenário de empreendimento com fontes de dados misturadas.

Vantagens: abordagem baseada em repositórios, suporta um amplo alcance de fontes de dados, possui documentação detalhada e material de tutorial bom.

Desvantagens: Possui suporte para o cliente limitado e a curva de aprendizagem é onerosa.

Evolutionary Technologies International (ETI)

http://www.eti.com/

Plataformas: *ETI*EXTRACT* executa em *mainframe IBM* compatível, *Microsoft NT*, *plataforma Sun Solaris*, ambas linguagens de programação *COBOL* e *C* e sistema de gerenciamento de dados *Sybase* e plataforma *UNIX*.

Funcionalidade de produto: *ETI*EXTRACT* é utilizada para gerar extração de dados, transformação e programa de migração de dados.

Vantagens: Possui uma infraestrutura que se adapta depressa a mudanças. Permite mudanças baseadas em descobertas de dados, demanda de usuários e fontes de dados. Mudanças para conversão e programas de interfaces de dados possuem execução muito rápida.

Desvantagens: Existe uma necessidade por uma ordem mais larga de plataformas. Enquanto *ETI*EXTRACT* executa mais amplamente desdobradas nas plataformas de *UNIX*, existe uma necessidade para uma porta *NT*.

Outras ferramentas:

Fabricante : *BMC Software Inc*.

Ferramenta: - *Change DataMove*2.2;

- *DataMove*2.2.

Fabricante: *Computer Associates International Inc.*

Ferramenta: *CA Data Acquisition Solutions*.

Fabricante: *coSORT/innovate routines international, Inc.*

Ferramenta: *coSORT v7 –SMP, sort/ETL Engine for UNIX and NT*

Fabricante: *Oracle Corporation*.

Ferramenta: *Oracle warehouse builder*.

Fabricante: *SAS Institute Inc*.

Ferramenta: *SAS™ Intelligent warehousing solution*.

Fabricante: *Taurus Software*.

Ferramenta: - *Bridgeware, version 9.8.*

- *Data Bridge™ open - ETL, version 2.06*.

3.2.6.5. Algumas ferramentas para Qualidade de dados:

Trillium Software

http: // www.trilliumsoft.com

Plataformas: O *trillium software system®* executa em *MVS,UNIX, Windows NT e 95, OS/2 e SAP.*

Funcionalidade do produto: O produto tem muita facilidade de uso. A configuração, manutenção e interface do produto são amigáveis e podem ser feitas por uma única pessoa.

Vantagens: Fornece um conjunto completo de bibliotecas de funções que podem ser incorporadas nas aplicações desenvolvidas pelo usuário. Agrupa registros além dos dados de nome e endereço.

Desvantagens: Tem limite para o número de elementos de dados usados no agrupamento. Não possui grande capacidade de análise.

Vality Technology Inc.
https://www.vality.com/

Plataformas: *Integrity Data Re-engineering Enviroment™ 3.3*. executa em *MVS, UNIX, Windows NT e AS/400.*

Funcionamento do produto: Capaz de consolidar dados provenientes de diferentes sistemas e efetuar reengenharia de dados em outros dados além de nome e endereço. Necessita de ajuda para configuração e manutenção por não possuir interface gráfica de usuário.

Vantagens: Alta qualidade de dados gerados. Analisa caractere a caractere de dados para detectar padrões, deixando pouco para ser corrigido manualmente. Simplifica e diminui o custo de migração de dados e utiliza reduzida equipe técnica, pela necessidade de poucos recursos de programação.

Desvantagens: Não faz a extração de dados, sendo necessário a aquisição de outros produtos para garantir esta exploração e também a qualidade dos metadados. Tem taxa de exatidão de dados conseguida inicialmente mais baixa que a de outros produtos, por encontrar qualquer tipo de dados.

Outras ferramentas:

Fabricante: *Ardent Software, Inc.*
Ferramenta*: Quality Manager.*

Fabricante: *Carleton*
Ferramenta: *Pure•View ™.*

Fabricante: *Firstlogic, Inc*.
Ferramenta*: i.d.Centric Data quality suite.*

Fabricante: *Qualitative Marketing software, Inc.*
Ferramenta: *Centrus ™ product suite*.

3.2.6.6. Considerações finais

Mesmo o objetivo do trabalho não sendo o estudo de ferramentas, observou-se que, quando realizando o processo de migração e a garantia de qualidade dos dados, é necessário o uso de ferramentas, e que muita embora novas ferramentas surjam diariamente, e freqüentemente surjam inovações, ainda não existe de fato uma ferramenta capaz de atender todos os aspectos necessários à migração de dados e qualidades de dados. Normalmente, para se obter isto, é necessário associar algumas ferramentas a outras para ser plenamente atendido.

CAPÍTULO IV

Análise das abordagens de migração de dados

4.1. Introdução

Este capítulo tem como propósito uma análise das quatro abordagens apresentadas no capítulo anterior para o processo de migração de dados de sistemas fontes para o *Data Warehouse*. Onde são feitas comparações e indicações de qual abordagem apresenta um conteúdo mais completo das atividades a serem cumpridas durante um processo de migração de dados.

4.2. Análise das abordagens

A primeira abordagem apresentada foi a de *Kimball* [KIM696], que embora tenha sido uma das primeiras abordagens formais ao processo de migração de dados, ainda permanece até os dias atuais como um eficiente método.

Neste trabalho, a abordagem de *Kimball* é apresentada em conjunto com o processo de limpeza de dados, haja vista que, a mesma apenas contempla o processo de extração e carga dos dados. Esta abordagem subdivide o processo de extração de dados em etapas que devem ser superadas para uma extração eficiente. Estas etapas são resumidas abaixo:.

Na **extração dos dados** devem ser realizadas leituras de dados legados - aqui foram ressaltados casos nos quais esta leitura por ser algo simples (para aplicações abertas e documentadas) e ou muito complexa (sistemas com estrutura proprietária). Nesta fase é importante a utilização de ferramentas para auxiliar os procedimentos. Durante a extração apesar de não ser tarefa simples, deve-se identificar dados novos dos já lidos e para tal algumas técnicas são utilizadas, para ajudar a diminuir o tempo de processamento. Incluir registros em banco já existente gera uma outra preocupação, que é solucionada com a adoção de uma aplicação de gerenciamento de chaves de identificação de registros, para que não haja conflitos de chaves na hora de se efetuar a

carga dos dados no DW. Segundo esta abordagem o local apropriado para que tudo isto ocorra é um ambiente intermediário de armazenamento de dados.

Na etapa da **limpeza de dados**, vários aspectos são levantados, tanto a respeito do local aonde a limpeza dos dados deve ocorrer e quando deixar dados sem limpeza. Estes questionamentos são respondidos de acordo com a visão de pesquisadores, como por exemplo: *Larry English* em [LARR99], que fabricam software de limpeza de dados para o *Data Warehouse*.

A limpeza de dados trata-se de um processo, no qual dados sujos provenientes de sistemas legados (fontes) são identificados. Para cada tipo de sujeira de dado é estabelecida uma regra de limpeza, que é registrada como metadados, que numa fase final do processo de limpeza de dados são utilizados para efetuar a limpeza dos dados propriamente dita. Procura-se apontar alguns tipos de sujeiras existentes nos sistemas legados, embora este tipo de identificação seja imprevisível, pelo fato da maioria dos dados sujos serem gerados durante a sua inserção nos sistemas fontes pelos operadores, mesmo assim, foram apresentados por *Larissa Moss* em [MOSS98], os tipos mais comuns já encontrados nestes sistemas.

Com o término das fases de extração e limpeza, onde os dados são transformados, outra fase é abordada: a **carga dos dados**, que é a efetivação da massa dos dados provenientes de sistemas legados para o *Data Warehouse*. Para esta abordagem algumas funcionalidades foram apontadas, tais como: o processo de carga ser o responsável em reconhecer os destinos dos dados e dar suporte para eliminar e recriar índices e proporcionar o particionamento físico de tabelas e índices. Em síntese a abordagem feita ao processo de carga consiste nos seguintes passos:

1º passo - antes da carga, devem ser criados os registros agregados, assim como, suas chaves artificiais para que não haja conflito com chaves para registro de nível básico.

2º passo - verificação da quantidade de campos dos registros de carga e os respectivos no Data *Warehouse*, garantindo compatibilidade necessária; além de ser fundamental a carga de dados com integridade referencial. Quanto aos registros que fracassam durante a carga, eles devem ser analisados, corrigidos e novamente carregados;

3º passo - após a carga, os índices afetados devem ser reconstruídos e deve ser assegurada a qualidade dos dados; para tal, é feita a utilização de relatórios e gráficos gerados a partir do *Data Warehouse* e comparados com os dados do sistema de origem.

4º passo - publicação de cada nova carga realizada, isto é, a carga no DW deve ser formalizada e publicada em manual ou em outras vias de comunicação para que o usuário final tenha disponíveis as novas alterações.

A segunda abordagem descrita neste trabalho foi feita por *Kathy Bohn* em [BOHN97], na qual a migração de dados consiste em elaborar um plano de conversão de dados e fazer especificações deste plano.

Esta abordagem é bastante completa, no sentido em que a elaboração de um plano de conversão leva em consideração desde os recursos disponíveis para o projeto, as linguagens de programação disponíveis e métodos de acessos recomendados. Para sistemas fontes localizados em diversas máquinas, defende-se que devem migrar para um esquema intermediário – uma área de estágio de dados. Quanto ao plano de conversão que estabelece, considera todas as fases seguintes:

1ª - como os dados migrarão dos sistemas legados para o esquema intermediário;

2ª - quais os procedimentos que a equipe de projeto deve ter durante a limpeza, transformação, integração, administração de chaves;

3ª - como os dados e metadados serão migrados do esquema intermediário para o servidor do *Data Warehouse*;

4ª - como a equipe deve fazer a carga e indexação dos dados do DW;

5ª - como a equipe deve assegurar a qualidade dos dados convertidos.

Com o estabelecimento do plano, especificações são criadas para que sejam alcançadas as fases do plano. As especificações de conversões que são geradas são na realidade metadados. Estes metadados possuem todo o mapeamento da fonte em relação ao banco designado, e também as regras de transformação a que os dados serão submetidos. Em forma de código de programação efetuam funções tais como: extração dos sistemas fontes, conversão dos esquemas intermediários para dados de carga, agregação dos dados de carga, migração dos dados de carga do esquema intermediário para o DW, carga dos dados e validação dos dados.

Estas funções estão minuciosamente detalhadas nesta abordagem. Desta forma, trata-se de uma abordagem com conteúdo completo e bem definido, embora seja necessário enfatizar que a mesma apresenta fundamentações muito fortes na anterior, chegando até ser mesmo totalmente iguais em algumas descrições de atividades.

A terceira abordagem é proposta por *John Shepherd* em [SHEP99] sendo uma estratégia em migração de dados, consistindo em um processo de dois passos:

1º passo - perfilamento de dados

2º passo - mapeamento dos dados.

A metodologia adotada por esta abordagem consiste na aquisição dos dados de sistemas legados bem definida, a principal preocupação é o entendimento completo dos dados fontes, perfilando e mapeando todas as modificações necessárias à migração dos dados. Em síntese, esta abordagem em relação as demais, funciona da seguinte forma:

- sintetiza no processo de perfilamento e mapeamento de dados, o que é executado nos processos de extração e limpeza das abordagens anteriores. Todos os procedimentos que ocorrem no processo de perfilamento são anotados como metadados que são utilizados no processamento de mapeamento, onde serão geradas as regras de limpeza e de transformação para a migração de dados.

Para esta abordagem também a utilização de ferramentas é necessária. A visão que se tem aqui é que, além de ferramentas que atendam o processo de extração, limpeza e carga, é necessário que sejam utilizadas também ferramentas que atendam as necessidades dos processos de perfilamento e mapeamento. Desta forma esta abordagem parece exigir mais recursos financeiros que as demais, pois exige mais utilização de ferramentas. Contudo, em sua essência é semelhante as outras abordagens, embora tenha um conteúdo que atenda apenas as técnicas de extração e limpeza dos dados. Devendo portanto, ser associada a outras técnicas para ser completa.

A quarta abordagem é proposta por *Janelle Hill* em [HILL98], esta abordagem trata da utilização de tecnologias de movimento de dados na preparação de dados para o *Data Warehouse*. As tecnologias são classificadas como: tecnologias mais simples, que são as que envolvem programas de transferências de arquivos; e tecnologias mais completas que envolvem ferramentas de transformação de dados.

Toda esta abordagem é feita analisando tecnologias existentes no mercado capazes de atender as necessidades de migração de dados. E demonstra que, embora existam técnicas e ferramentas disponíveis no mercado, nem todos os requisitos da migração de dados são contemplados, principalmente quando se trata da utilização de ferramentas como soluções para aplicações, como por exemplo, captura de dados modificados.

Esta abordagem em síntese, pretende indicar tecnologias apropriadas para que a migração de dados ocorra.

Observou-se que em todas as abordagens a garantia de qualidade de dados é um ponto em comum. E que a qualidade de dados deve ser obtida durante os processos de extração e limpeza, mesmo antes da carga ser feita no *Data Warehouse*, garantindo desta forma, maior eficiência no processo de migração, que é de vital importância a um projeto de *Data Warehouse*, por isso, deve ser bem executado, e associado ao processo de qualidade de dados, além do fato de que a utilização de técnicas e métodos eficientes se faz necessário, assim como, o uso de ferramentas de última geração.

4.3. Considerações e limitações sobre os trabalhos pesquisados

No capítulo anterior e nas seções anteriores deste Capítulo foram detalhadas quatro abordagens de processo de migração de dados para o DW. Dentre estas, observou-se que o trabalho de [BOHN97] é o mais completo, abordando todas as fases do processo de migração de forma bastante enfática e precisa. Percebe-se também que este trabalho assemelha-se muito ao de [KIM696] em essência de conteúdo.

A Tabela 2, apresenta de forma resumida e comparativa as metodologias estudadas e os critérios de análise considerados.

Neste momento é oportuno relembrar que existe a falta de uma metodologia no meio acadêmico adequada ao contexto de migração de dados da tecnologia DW.

CARACTERÍSTICAS	KIM696	SHEP99	HILL98	BOHN97
Ser completa	**Não**	**Não**	**Não**	**Sim**
Atividades	**Extração de dados Carga dos Dados Assegura qualidade de dados Ferramentas**	**Perfilamento de dados Mapeamento de dados Ferramentas**	**Identificação dos dados Aquisição dos dados Limpeza dos dados Transformação dos dados Atualização do DW Ferramentas**	**Plano de conversão para assuntos desde a migração de dados, limpeza, transformação e integração, carga, administração de chaves, qualidade dos dados convertidos. Especificações de conversão, sendo aprovada pelo usuário Ferramentas**
Detalhe das atividades	**Bem detalhada**	**Muito pouco detalhe**	**Pouco detalhe**	**Bem detalhada**
Experimentação de ferramentas e tecnologias para executar as atividades.	**Enfatiza**	**Enfatiza**	**Enfatiza muito**	**Enfatiza**

TABELA 2 - RESUMO DO ESTUDO DAS ABORDAGENS DE MIGRAÇÃO DE DADOS PARA DW

Fazendo a comparação destes requisitos com o trabalho de cada autor, resumido na Tabela 2, observam-se as seguintes particularidades:

Somente [BOHN97] apresenta uma abordagem de migração de dados completa em relação as atividades sugeridas , a proposta feita por [SHEP99] é bastante resumida.

A proposta [KIM696] apresenta conteúdo incompleto tratando apenas as atividades superficialmente, deixando algumas atividades, tais como transformação de dados sem nenhum tratamento.

A proposta feita por [HILL98] enfatiza mais a utilização de ferramentas para resolver as atividades do processo de migração de dados, do que descrição de técnicas para solucioná-los.

No contexto desta dissertação, considera-se que uma metodologia de migração de dados para DW, deve satisfazer no mínimo os seguintes requisitos estabelecidos na fase do projeto e desenvolvimento da área de organização de dados do ciclo de desenvolvimento de DW, já descrito nesta dissertação por [KIM98L]:

1º requisito - ser completa – abordando desde o inicio até a conclusão das atividades da fase do projeto de desenvolvimento.

A proposta de [BOHN97] é a mais completa em relação as atividades fundamentais de extração, transformação e carga de dados, e através de um plano de conversão de dados executa todas as atividades necessárias a um bom processo de migração de dados.

A proposta de [KIM696] é incompleta quando aborda a limpeza dos dados , esta proposta denominada arquitetura de extração de dados compreende 13 passos que embora muito detalhados contemplam apenas a extração e carga dos dados da tecnologia DW.

A proposta de [SHEP99] trata de um planejamento com a finalidade de gerar mapas de transformações para facilitar a captura de requisitos de limpeza e de transformações , não especificando nenhuma atividade de extração, transformação e carga, sendo incompleto em atender as atividades que devem ser desenvolvidas para a boa execução do desenvolvimento da área de organização de dados.

A proposta de [HILL98] apresenta estratégias para preparar dados para o *Data Warehouse*, mas é muito resumida quando descreve estas, tornando-se incompleta e insatisfatória a utilização desta proposta.

2º Requisito - Ser detalhada - todas as atividades devem ser bem detalhadas, facilitando a execução das diversas etapas.

Observou-se o seguinte para este requisito:

A abordagem de [KIM696] apesar de ser bastante detalhada, aborda somente a extração, carga dos dados e a garantia de qualidade de dados, deixando as atividades de limpeza de dados em aberto.

A abordagem de [HILL98] detalha pouco o processo de migração de dados, chegando a comentar rapidamente as estratégias utilizadas para este processo.

A abordagem de [BOHN97] detalha bem as fases correspondentes às atividades. Das abordagens estudadas, essa é a que melhor atende ao requisito de detalhamento do processo de migração de dados para *Data Warehouse.*

3º Requisito - possibilitar a experimentação de ferramentas e tecnologias para executar as atividades.

Para este requisito observou-se o seguinte:

Todas as abordagens contemplam este requisito, principalmente a abordagem de [HILL98].

4.4. Conclusão sobre as propostas pesquisadas

As abordagens apresentadas neste capítulo possuem um objetivo comum: a migração de dados fontes para o *Data Warehouse*. Estas abordagens são bastantes semelhantes e algumas vezes complementares. Uma das abordagens que se destaca por ser mais detalhada, e por isto ser a mais completa é a de *Kathy Bohn* em [BOHN97].

Embora a proposta feita por [BOHN97] seja a melhor em relação aos requisitos formulados, deixa a desejar sob alguns aspectos para aplicá-la, uma vez que os termos técnicos e instruções que a mesma utiliza, são mais adequados para especialistas no assunto e que possuam conhecimentos em técnicas de migração em *Data Warehouse*, o que em meios acadêmicos, impossibilitaria implementações completas sem a ajuda de profissionais da área.

4.4.1. Um exemplo de passos para migrar dados para o DW

Este exemplo não possui especificações de dados e nem tampouco especificações técnicas sobre qualquer órgão ou empresa, serve apenas para demonstrar quais passos são utilizados, de acordo com a visão e entendimentos teóricos adquiridos pelo estudo desenvolvido neste trabalho, durante o processo de migração de dados para o *Data Warehouse.* Não se levará em consideração a complexidade do banco de dados, uma vez que isto só seria possível se estivesse sendo trabalhado dados concretos.

Hipoteticamente será executado um processo de migração de dados de sistemas fontes para o *Data Warehouse*, utilizando as abordagens analisadas.

Para efetuar este processo, os passos adotados são:

1. **Criar um plano de conversão** – seguindo especificações em [BOHN97];
2. **Criar as especificações de conversão** – seguindo especificações em [BOHN97];
3. **Extrair os dados** - seguindo os passos adotados em [KIM696], que compreendem: leitura dos dados legados, identificação de novos dados, generalização de chaves, combinação de registros de diversas fontes;
4. **Limpar os dados** – seguindo os requerimentos de limpeza identificados no plano de conversão em [BOHN97], gerando regras de transformações elaboradas a partir do tipo de sujeira identificada, como em [MOSS98]; o que consiste basicamente em:identificar dados sujos e processar a limpeza dos dados utilizando ferramentas.
5. **Carregar os dados** – utilizando as funcionalidades propostas por [KIM696], que compreendem: criar imagens de registro carga, criar agregações, generalizar chaves para registros de agregação, carga de registros com integridade referencial, tratamento de registros rejeitados, construir índices e assegurar qualidade dos dados.

Na possibilidade de se idealizar um modelo para migração de dados de DW, o adequado é utilizar os passos adotados na segunda abordagem [BOHN97], observando procedimentos adotados pela primeira abordagem em [KIM696] e verificando as tecnologias disponíveis pela quarta abordagem em [HILL98]. No entanto, uma indicação de uma abordagem ideal é algo muito complexo, tanto pelo fato destas abordagens possuírem conteúdos muitos semelhantes, como pelo fato de que, a complexidade dos dados na fonte serem decisivos no processo a ser adotado.

Tomando isto como base, é mais aconselhável combinar estas técnicas propostas e associar a elas as técnicas de qualidade de dados e ferramentas adequadas para uma migração de dados eficiente.

CAPÍTULO V

Conclusões e Recomendações

5.1. Conclusão

Com a pesquisa realizada em *Data Warehouse*, verificou-se que a utilização de Sistemas de Apoio à Decisão (SAD), tem sido intensiva por parte das organizações, para poderem posicionar-se mais vantajosamente perante as outras, garantindo até mesmo a sua própria evolução e manutenção no mercado. Neste contexto verificou-se que *Data Warehouse* e *Data Mart*, auxiliam administradores de primeira linha de negócios (gerentes, chefes, etc), fornecendo-lhes informações capazes de identificar tendências e padrões, para que sejam estabelecidas táticas e estratégias de negócios.

O presente trabalho apresenta um embasamento conceitual na área de *Data Warehouse*, onde conceitos básicos correlatados com esta tecnologia são apresentados. Além de conceitos sobre a própria tecnologia, tais como: características, arquiteturas, topologias que são abordados dentro do ambiente do *Data Warehouse*. Ambiente no qual os seus componentes foram conhecidos e conceituados. Dentre estes componentes, um mereceu destaque o *Back_End*, por conter a Área de estágio de dados, onde os processos de extração, limpeza e carga (migração dos dados) e qualidade dos dados ocorre.

Foram estudadas abordagens de migração de dados e qualidade de dados, além de ter sido citadas ferramentas capazes de atender a estas necessidades. Um estudo das abordagens apresentadas foi feito a fim de indicar entre estas a mais completa conceitualmente e comparações foram realizadas. Ficou claro que os processos de migração de dados são de fundamental importância no projeto de um *Data Warehouse*, pois são estes processos que vão proporcionar os dados finais que deverão povoar o repositório. E que o ideal seria a existência de uma metodologia capaz de contemplar todas as fases de atividades do processo de migração de dados, e que nesta metodologia a qualidade de dados fosse garantida durante a realização de todo o processo, evitando que procedimentos já executados fossem refeitos, desta forma, não seriam onerados

custos e tempo. A etapa de migração requer mais tempo e recursos humanos do que as demais fases do desenvolvimento do DW, e trata-se também da mais importante.

Todo o conteúdo coletado e produzido mostra que *Data Warehouse*, além de ser fundamental a organizações, deve ser adotado por aquelas que pretendem se destacar no mercado globalizado. E que se trata de uma tecnologia em plena evolução, tanto na área de técnicas e métodos de desenvolvimento, como em descobertas e aprimoramento de ferramentas.

5.2. Recomendações

Algumas pesquisas podem ser realizadas nesta área tais como:

- Estudo comparativo de várias ferramentas de *Data Warehouse* para migração de dados e também para qualidade de dados a partir de um estudo de caso, estabelecendo o que cada ferramenta atende das técnicas exigidas para migração e estabelecendo as exigências necessárias para a migração de dados eficiente.

- Estudo aprofundado sobre a utilização de metadados no processo de migração de dados.

- Estudo conceitual aprofundado do processo de migração de dados e ferramentas disponíveis voltados para o uso de *Data Warehouse* na Web.

ANEXO 1 - *EXPLORATION WAREHOUSE*

A1.1. *Exploration Warehouse*

Segundo [HALL99]: O *Exploration Warehouse* é uma arquitetura bastante nova que começou a surgir durante o ano de 1998. Esta arquitetura não é uma substituição para arquiteturas de *Data Warehouse* mais tradicionais. É significativa para operar como um suplemento a *Data Warehouse*, *Data Warehouse* de empreendimento, e arquiteturas de *Data Mart*. O *Exploration Warehouse* consiste em duas formas:

Warehouse protótipo - que provê um desenvolvimento de ambiente de banco de dados *RAD*(Rapid Application Development) para os projetistas e desenvolvedores de *Data Warehouse*.

Exploration Warehouse - que proporciona para os usuários finais um ambiente de *Data Warehouse* que suporta consultas ad hoc e "qualquer" análise de dados

A1.1.1 – *Warehouse* protótipo

O *Exploration Warehouse* usado no papel de prototipagem serve como uma plataforma de desenvolvimento interativa, na qual o desenvolvedor pode usar no projeto inicial do *Data Warehouse* e estágios de desenvolvimento para várias tarefas, incluindo:

- Avaliação de integridade de dados fonte e necessidades de transformação de dados para *Data Warehouse*;
- Junção da aplicação e requisitos de usuário final ;
- Criação de um modelo de dados inicial;
- Construção de uma aplicação piloto para testar manualmente.

Desta maneira, como mostrado na Figura A1_1, os projetistas extraem e combinam dados de fontes discrepantes, criando um armazém de protótipo que eles podem experimentar então com os futuros usuários finais.

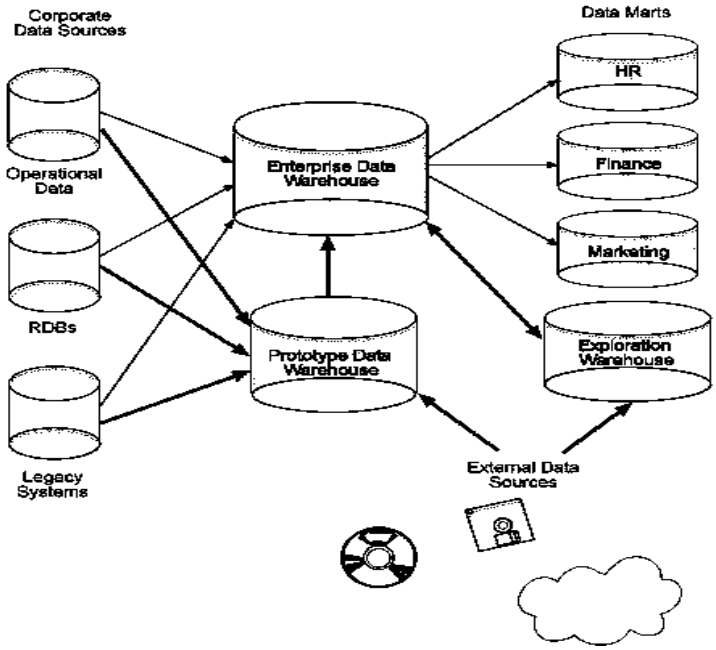

Figura A1_1 – Função do *Exploration Warehouse*

Baseado em especificações iniciais colecionadas durante construção do armazém de protótipo e em realimentação adquirida de sugestões de usuário final, projetistas usam o *warehouse* protótipo como um veículo para sessões de testes temporárias, o administrando para determinar as exigências do *warehouse* de produção. Este projeto específico de *warehouse* pode se desenvolver eventualmente no *Warehouse* de produção ou *Data Mart*, ou pode servir como um *warehouse* de prova de conceito. Indiferentemente, são aplicadas a experiência e especificações ganhas do esforço de *warehouse* de protótipo, então para construir o *Data Warehouse* de produção. Alternativamente, em casos onde uma organização já tem em uso o *Data Warehouse*, o *warehouse* protótipo é usado para ajudar a atualização e modificação dos *warehouses* de produção existentes.

Desta forma, o benefício principal do *warehouse* protótipo é que provê um ambiente *warehouse RAD* que os projetistas de *Data Warehouse* podem usar para acelerar projetos de armazém de dados e implementação, rapidamente por desenvolvimento interativo e testado com futuros usuários finais.

A1.1.2 - *Warehouse* de exploração

O *warehouse* de exploração é um *warehouse* ou *Data Mart* especificamente usado no papel analítico por usuários finais que querem administrar verdadeiramente consultas *ad hoc* e "qualquer" análise exploratória. O *warehouse* de exploração é engrenado principalmente para os analistas mais avançados e usuários de alto nível dentro da organização, cujas necessidades de análise de dados vão além dos relatórios prontos e capacidades de análise requeridas por usuários de *warehouses* mais populares. Estes dominam as necessidades dos usuários rapidamente e acesso pronto para muitos dados atuais nos quais eles freqüentemente podem emitir questões complexas e execuções longas de consultas freqüentemente associadas com *OLAP*, *data mining*, e outras aplicações de inteligências empresariais.

Embora, o *Exploration Warehouse* primário de alimento de dados é o *Data Warehouse* de uma companhia de empreendimentos (e até às vezes *ODS*), o *warehouse* de exploração normalmente é mantido separado destes armazéns operacionais, e *Data Mart* são usados para evitar taxar muitos recursos de sistemas e se aprofundar mais nestes sistemas (Figura A1.1.1). Além disso, os usuários de *warehouse* de exploração freqüentemente precisam integrar dados externos (por exemplo, comercializados, *census/demographic*, Internet, ou outros dados adquiridos de provedores de dados de outros fabricantes) para administrar as suas análises.

A1.1.3 - Exigências de bd para *warehouse* de exploração e de Protótipo

As exigências do *warehouse* protótipo e de exploração são tais que provêem funcionalidades ótimas, eles exigem para o uso de um banco de dados (bd) que apresente várias capacidades importantes e características. Portanto, para um *warehouse* protótipo ser efetivo e oferecer verdadeiramente um ambiente de desenvolvimento *warehouse RAD*, é exigido um banco de dados que provê:

- Rápida e fácil capacidade de projeto e construção, que eliminam a necessidade para criar e manter estruturas de dados explícitas;
- A habilidade para rápida e fácil carga de dados de uma variedade de fontes, inclusive bancos de dados, flat files, sistemas legados, e outras fontes de dados de terceiros (por exemplo, geográfico, marketing, e dados de Internet);
- A habilidade para construir tabelas "flutuantes" que são indexadas completamente em ordem, para apoiar processo analítico imprevisto;
- A habilidade para usuários imediatamente analisar dados executados uma vez e carregados no banco de dados .
- A habilidade para somar dados adicionais qualquer hora em qualquer tempo, sem a exigência que seja primeiro reestruturado ou reprojetado o banco de dados.

Para um *warehouse* de exploração ser efetivo e prover verdadeiramente um *warehouse* analítico *ad hoc*, é requerido que um banco de dados proveja as mesmas características como requeridas para o *warehouse* de protótipo. Porém, o banco de dados de *warehouse* de exploração também deve poder dirigir questões muito complexas que freqüentemente não são questões ótimas porque eles freqüentemente são escritos por usuários finais que possuem um conhecimento limitado de SQL. O banco de dados de *warehouse* de exploração deve também prover capacidades de administração automatizadas para reduzir, o máximo possível a necessidade de usuários finais ter que chamar a equipe de desenvolvimento e suporte.

O *warehouse* de exploração que usa tecnologia de banco de dados tradicional pode ser construído; porém, é melhor usar um banco de dados que ajude a eliminar os *overheads* associados com bancos de dados tradicionais.

Dois produtos foram projetados especialmente para *exploration warehouse*:

Fabricante: *Sand Technology Systems International*.
Produto: *Nucleus Exploration Warehouse and Nucleus Exploration Mart.*

Fabricante: *Digital Archaeology, Inc.*
Produto: *Digital Archaeology Discovery Suite.*

ANEXO 2 - Arquitetura *Data Warehouse Bus*

A arquitetura *DW bus* combina dois conceitos chaves (*Data Mart* e Modelos dimensionais). [KIM1198].

A2. 1 - *Data Mart* e Modelagem Dimensional

Dois princípios chaves para guiar projetos de *Data Warehouse*:

1. Separar as arquiteturas. Separar o *back end* e o *front end*. O *back end* é onde se extrai, transforma, e carrega os dados, e o *front end* é onde os dados estão disponíveis para apresentação.

2. Construir *Data Mart* orientados a apresentação em volta de modelos dimensionais, não modelos *ER*.

A figura A2.1, mostra um corte horizontal típico para um ambiente de *Data Warehouse* global. À extrema esquerda se vê os sistemas legados tradicionais orientados à transação. A responsabilidade do DW começa quando os dados são extraídos dos sistemas legados e entram na *data staging area* (área de organização dos dados).

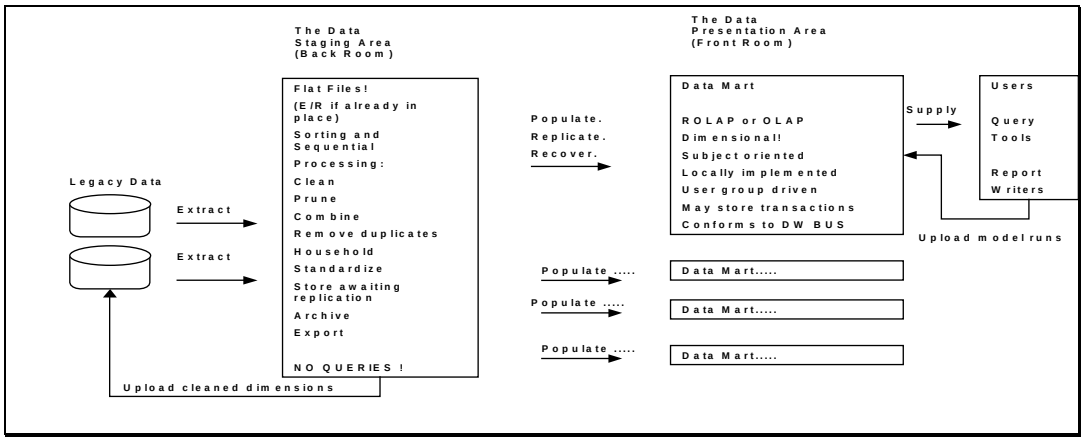

Figura A2-1 - O *Data Warehouse* de empreendimento, mostrando a área de organização dos dados e a área de apresentação dos dados.

Área de Organização dos dados

A área de organização dos dados é a completa operação *back end* para o *Data Warehouse* no qual os dados são limpos, selecionados, combinados, ordenados, observados, chaves são adicionadas, dados duplicados são removidos, agregados, arquivados, e exportados. Os dados que chegam na área de organização dos dados estão freqüentemente sujos, malformados, e em um formato de *flat-file*. Na melhor das hipóteses os dados chegam em terceira forma normal prístina, mas isso é raro. Quando a limpeza é finalizada e os dados reestruturados, pode-se deixar isto na forma flat file ou pode-se armazenar isto em terceira forma normal. A área de organização dos dados é dominada por flat files, ordenação simples, e processamento seqüencial.

Os requisitos arquiteturais chaves da área de organização de dados devem estar fora dos limites dos usuários finais e de todas as formas de pesquisas deles. A principal razão é o fato da equipe de projeto do DW, não ter que se distrair provendo disponibilidade de dados, índices, agregações, integração síncrona por áreas de assunto, e especialmente segurança ao nível de usuário.

Área de Apresentação

A área de apresentação é a completa operação de *front end* para o DW. A área de apresentação está organizada e disponível a toda hora para investigação de usuários finais. Todas as formas de investigações de usuários finais são concedidas pela área de apresentação incluindo: consulta *ad hoc*, *drilling dow*, relatórios, e *data mining*.

A área de apresentação está subdividida em áreas de assunto que são chamadas *Data Marts*. Cada *Data Mart* é completamente organizado através de apresentação efetiva - modelos dimensionais. Apresentando toda a investigação e atividades de análise que incluem consulta *ad hoc*, geração de relatórios, ferramentas de análise final superior, e *data mining*. Todos os modelos dimensionais em todos os DM parecem um pouco semelhantes, e este *suite* de modelos dimensionais tem que compartilhar as dimensões chaves do empreendimento. Estas dimensões são denominadas dimensões conformadas.

A2. 2 - Plugando os *Data Marts* na arquitetura de *Data Warehouse bus*.

A figura A2.2, mostra como prender vários *Data Marts* independentes junto com dimensões conformadas (conjunto constantemente definido de dimensões que todos os *Data Marts*, que desejam referir-se a estas entidades comuns, devem usar). Dimensões conformadas incluem coisas tipicamente como: calendário (tempo), cliente, produto, localização, e organização. Tem que se levar em consideração fatos conformados que envolvem qualquer medida que existe em mais de um *Data Mart*. Para conformar várias instâncias de um assunto em comum, as definições técnicas de cada instância devem ser da mesma forma que o elemento de dado da instância para que possam ser comparados e adicionados. Se não podem ser conformadas duas versões do elemento de dado, eles devem ser etiquetados diferentemente, desse modo, os usuários não compararão ou somarão versões.

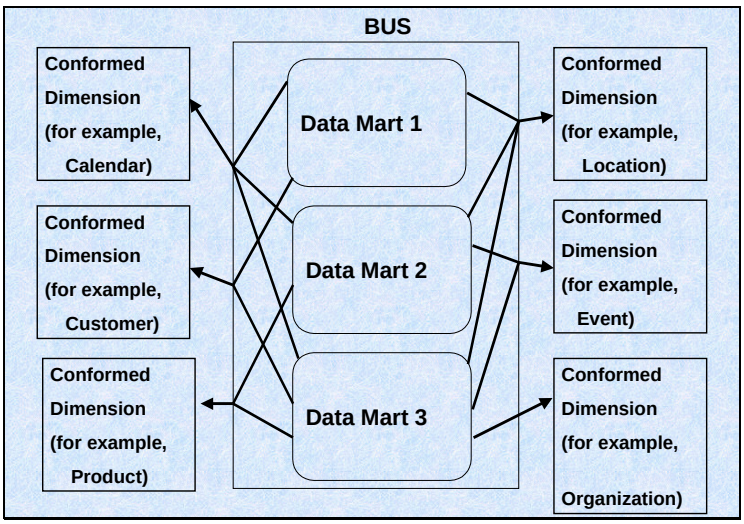

Figure A2.2. A arquitetura de *Data Warehouse bus*, mostrando uma série de *Data Marts* independentes que conectam as dimensões conformadas do empreendimento.

Suponha que o *DW bus* é igual ao barramento do computador. O barramento no computador é uma especificação de interface padrão que permite plugar em um *CD-ROM*, uma unidade de disco, ou qualquer número de cartões de propósito especiais. Por causa do padrão de barramento, estes periféricos trabalham juntos com harmonia, embora eles sejam fabricados em momentos diferentes por vendedores diferentes. Da mesma maneira, o *DW bus* é um padrão que permite implementar *Data Marts* separados através de grupos diferentes na empresa em momentos diferentes. Aderindo ao padrão (dimensões conformadas), os DM separados podem ser plugados junto. Eles podem até mesmo compartilhar dados proveitosos em um relatório *drill-across* porque o topo da fileira do relatório significará a mesma coisa para cada um dos DM.

Separando a arquitetura global do DW completamente, a área de organização de dados e a área de apresentação executam funções distintamente diferentes. A arquitetura bus que define a área de apresentação de dados confia na semelhança previsível dos modelos dimensionais para deixar o empreendimento enlaçar todos os *Data Marts* juntos. Esta abordagem é uma arquitetura para endereçar todos os novos bravos requisitos na indústria de DW.

A2. 3 – Novos requisitos na indústria de DW

Os novos requisitos para a indústria de *Data Warehouse* são [KIM1098]:

Desenvolvimento incremental Descentralizado - Dentro de uma organização, departamentos e divisões vão criar os seus próprios mini *Data Warehouse* para responder perguntas de negócio urgentes. Logo, deve ser criada uma estrutura e disciplinar este processo de desenvolvimento, possibilitando que uma equipe em particular de *Data Mart* seja capaz de implementar sem saber o que as outras equipes estão fazendo em detalhes.

Antecipação de mudança contínua como necessidades de negócio e evolução de fontes de dados disponíveis - Os requisitos que mantém o esquema de banco de dados constante é extremamente importante. Se esta meta não é assegurada aplicações

existentes continuarão trabalhando, até mesmo depois de adicionados novos dados para o ambiente.

Desenvolvimento rápido - A exigência para construir as partes do *Data Warehouse* rapidamente provavelmente designa a primeira exigência para desenvolvimento descentralizado e incremental. Para DW centralizados é difícil imaginar um desenvolvimento rápido, pelo fato de que é necessário que ele esteja completo para ser utilizado e isto certamente demanda tempo. Além disto, desenvolvimento rápido também significa que as técnicas para construir as partes do *Data Warehouse* são bem compreendidas, previsíveis, e simples. Se todas as partes do DW tiverem a mesma visão e estrutura torna-se possível carregar estas partes, indexá-las para desempenho, selecionar ferramentas para acessá-las e consultá-las.

Drill dowm sem emendas para os mais baixos dados atômicos possíveis - Drilling down não é nada além que adicionar um topo de fila para um relatório existente. Em uma arquitetura de dimensão conformada, estes topos de filas são conhecidos para ser acessíveis nas dimensões e possuirão um significado consistente quando descendendo de tabelas fato mais agregadas para menos agregadas. Os dados mais atômicos são os mais naturalmente dados dimensionais, porque a maioria de atributos valorados simples existe para cada registro de tabela fato neste nível. Os dados atômicos são necessários na maioria dos DM.

As partes (Data Mart) somando o todo (Data Warehouse) - A exigência que o DW está composto de nada mais e nada menos que a soma dos DM é em grande parte uma conseqüência das exigências prévias. As áreas de assunto separadas vão ser implementadas em um modo distribuído. Cada *Data Mart* vai conter seus dados atômicos subjacentes. Os dados de medida numéricos não devem ser duplicados em múltiplos lugares ao redor do empreendimento; estes dados são fortemente a maior parte de qualquer *Data Mart*.

As partes (Data Mart) implementadas em tecnologias diversas, incompatíveis - Os *Data Mart's* podem ser incompatíveis ao mais baixo nível de hardware e software, porque o

hardware e software não se comunicam diretamente. Executando consultas separadas para cada *Data Mart* (usando diversas passagens denominado *SQL* e seu equivalente para bancos de dados *OLAP*), se combina um conjunto de respostas em uma camada de aplicação de alto-nível. Esta abordagem também tem o benefício significativo no qual as consultas separadas evitam um *host* de problemas lógicos complexos associados com dificuldade para juntar tabelas fato com diferentes cardinalidades.

2437 disponibilidade. A exigência para 2437 disponibilidade se refere à área de apresentação de dados. O primeiro passo para alcançar 2437 disponibilidade é implementar a área de organização dos dados em uma máquina separada ou em um processo separado da área de apresentação de dados. A produção final da área de organização dos dados é um conjunto de arquivos de carga para a área de apresentação de dados. Porém, carregar e indexar estes arquivos no banco de dados de apresentação final pode ser um processo longo que leva o de banco de dados de apresentação *offline*. Para evitar *offline* por longos períodos, pode-se usar uma estratégia mencionada de arquivo. A carga de banco de dados de cada manhã entra em um "arquivo de tempo" que começa com uma cópia completa do banco de dados de apresentação normal. Quando o arquivo de tempo estiver carregado e indexado, o sistema vai a *offline* por alguns segundos, enquanto a tabela de banco de dados de produção corrente é renomeada, e o arquivo de tempo é nomeado como a tabela de banco de dados de produção.

Publicando resultados de Data Warehouse em todos lugares, preferentemente na Internet - Embora arquitetos de dados e alguns consultores da indústria de Serviços de Informação individuais não estão completamente convencidos que a abordagem dimensional e consultas simples são requeridas, os vendedores de ferramentas estão quase todos provendo interfaces de usuário habilitadas à rede. A maioria dos donos de *Data Warehouse* se achará apresentando os seus Datas *Warehouse*s sobre *intranets* ou até mesmo a Internet, se eles planejam isto ou não.

Segurança para resultados de Data Warehouse em todos os lugares, especialmente na Internet - O lado ruim de usar o meio de transporte onipresente da *Internet* é a

exposição a problemas de segurança. O dono de armazém de dados é especialmente vulnerável por causa da sensibilidade de muito dos dados subjacentes, e por causa, ironicamente do sucesso do armazém, publicando os dados efetivamente para todos os usuários finais.

Resposta instantânea próxima para todos os pedidos - Tempos de resposta melhorados para consultas de usuários finais incluem: disciplina para usar banco de dados simples, previsível estrutura e aumento do uso de índices de banco de dados com agregações de banco de dados, e uso de diversas passagens *SQL* em vez de *SQL* complexo monolítico. Todas estas abordagens usam a aproximação do modelo dimensional pesadamente, e há um corpo crescente de experiência e tecnologia nestas áreas, baseado em suposições dimensionais.

Facilidade de uso, especialmente para pessoas que usam pouco computador - A exigência final da lista é de fato a exigência mais importante. Simplesmente usuários finais não usarão algo que é de difícil uso. Ou talvez um subconjunto minúsculo de entusiastas técnicos usará algo complicado. Ou talvez o sacerdócio de desenvolvedores de aplicação que são os únicos verdadeiros usuários acabe. Facilidade de uso é o principio de tudo. Interfaces de usuários final, que são reconhecíveis, memoráveis, alto desempenho, e estão baseadas em modelos que podem ser invocados ou modificados em um simples click de botão.

GLOSSÁRIO

API (Application Programming Interface) – interface de programas aplicativos – conjunto de rotinas utilizadas para controlar a execução de procedimentos por parte do sistema operacional do computador.

AUDITORIA - Exame rigoroso de equipamentos, programas, atividades e procedimentos para determinar com que eficiência o sistema como um todo está funcionando, principalmente na garantia da integridade e da segurança de dados.

BUFFERS – Região da memória reservada para ser utilizada como repositório intermediário de dados mantidos temporariamente.

CASE (Computer aided software engineering) – softwares que permitem o uso do computador em todas as fases do desenvolvimento de sistemas, desde o planejamento e a modelagem, até a codificação e a documentação.

COMMIT – Condição feita pelo programador ao SGBD, sinalizando que atualizações devem ocorrer no banco de dados.

DBMS (Database Management System) – O mesmo que SGBD.

DICE - extração de um subcubo ou a intersecção de vários slices.

DML (Data Manipulation Language) – Linguagem de manipulação de dados - Usada para inserir dados em um banco de dados e para permitir a atualização deste.

DRILL DOWN – Começar em um menu, diretório ou página da Web de nível mais alto e passar para menus, diretórios ou páginas vinculadas, até o arquivo, página, comando de menu ou outro item procurado ser encontrado. Em resumo, a partir de um nível de dado mais alto, são obtidos dados mais detalhados. Ou ainda, Aumentar o nível de detalhes de uma consulta ou relatório, adicionando-lhes novas linhas de cabeçalho provenientes

de tabelas dimensão. O verdadeiro drill down deve permitir utilizar qualquer atributo disponível nas tabelas dimensão.

DRILL UP – o inverso de drill down, ou melhor, os dados são apresentados em um nível mais elevado a partir de um nível mais detalhado.

ETI (Extraction/ Transformation Integration) – designa o processo e ferramentas que atendem a Extração, Transformação e Integração de dados.

ETT (Extraction/ Transformation Transport) – designa o processo e ferramentas que atendem a Extração, Transformação e Transporte de dados.

FLAT FILE – Flat files são arquivos seqüenciais com caracteres ou campos de controle que permitem incluir múltiplos registros lógicos de informação em um único arquivo. Esses arquivos foram muito utilizados no tempo em que o espaço de armazenamento era uma séria restrição aos sistemas computacionais, sendo uma forma conveniente para armazenar diferentes tipos de arquivos lógicos juntos, em um mesmo arquivo físico. Ou ainda, flat file é um arquivo formado de registros do mesmo tipo, sem que exista uma estrutura interna definindo as relações entre os registros.

MOLAP (Multidimensional OLAP) - também denominado Banco de Dados Multidimensional *(*Multidimensional database - MDDB), constitui-se de um conjunto de interfaces de usuário, aplicações e banco de dados, com tecnologia proprietária, que possui características eminentemente dimensionais. Bancos de dados multidimensionais (MDDB) armazenam seus dados em um cubo de "n" dimensões e adiciona tempo as dimensões.

ODBC (Open Database Connectivity) – uma interface que forncece uma linguagem comum para se ter acesso a um banco de dados de uma rede.

OLAP (On-line Analytic Processing) - Processamento Analítico On-Line, constitui-se de

todas as atividades gerais de consulta e apresentação de dados numéricos e textos provenientes do DW, assim como as formas específicas de consulta e apresentação que são exemplificadas por uma grande quantidade de ferramentas OLAP através do acesso multidimensional aos dados.

OLTP (On-line Transaction Processing) – Processamento de Transações On-Line, utilizado para processar transações assim que o computador as recebe, atualizando imediatamente os arquivos mestres de um SGBD.

OVERHEAD – Atividades ou informações que oferecem suporte a um processo de computação, mas não fazem parte intríseca da operação ou dos dados.

PIVOT - é o ângulo pelo qual os dados são vistos ou trocados. É a modificação da posição das dimensões em um gráfico ou troca de linhas por colunas em uma tabela.

RAD – Método de construção de sistemas de computador em que o sistema é programado e implementado em segmentos, não aguardando o projeto inteiro está concluído para dar início à implementação. Utiliza ferramentas CASE e programação visual.

REPLICAÇÃO – Processo que consiste em copiar um banco de dados (ou parte dele) para outros locais.

ROLAP (Relational OLAP) - Relacional OLAP, constitui-se de um conjunto de interfaces de usuário e aplicações que dá ao banco de dados relacionais características dimensionais.

ROLL UP – mesmo que DRILL UP.

SGBD – Sistema de Gerenciamento de Banco de Dados. Interface de software entre o banco de dados e o usuário.

SLICE - extração de informação sumarizada de um cubo de dados, a partir do valor de uma dimensão. O uso integrado dos conceitos “slice e dice” permite rotacionar os lados de um cubo de dados(dimensões) em qualquer sentido, possibilitando a combinação de quaisquer dimensões e a obtenção de informações correspondentes sobre vários enfoques.

SNAPSHOT – Instantâneos. Cópia da memória principal em determinado instante, enviada para o disco rígido.

STOVEPIPE – Nova nomeclatura para as antigas ilhas - sistemas operacionais sem integração e compartilhamento de dados com outros sistemas.

REFERÊNCIAS BIBLIOGRÁFICAS

[BER97] Berson, Alex & Smith, Stephen J. – **Data Warehousing, Data Mining & OLAP** – Nova York: McGraw-Hill, 1997.

[BOHN97] Kathy Bohn. **Converting data for *warehouse*s**. DBMS Magazine. Jun97. Capturado em 15 de Agosto de 2000. Online. Disponível na internet : http://www.dbmsmag.com/9706d15.html.

[BOKU98] Bokun, Michele & Taglienti, Carmem. **Incremental *Data Warehouse* updates: approaches and strategies for capturing changed data**. DM Review Magazine, May; 1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.dmreview.com/master.cfm?NavID=55&EdID=609.

[BON98] Bontempo, Charles & Zagelow, George. **The IBM - Data Warehouse Architecture**. Communications of the ACM, 41 (9): 38-48. Sept; 1998.

[DYCH] Dyché, Jill. **Scoping your *Data Mart* implementation**. DBMS magazine, August 1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet http://www.dbmsmag.com/9808d13.html.

[FURL94] Furlan, José Davi et ali. **Sistemas de informação executiva = EIS – Executive Information Systems: como integrar os executivos ao sistema informacional das empresas....** .São Paulo: Makron Books, 1994.]

[GAR98] Gardner, Stephen R. **Building the Data Warehouse**. Communications of the ACM, New York, v. 41, n. 9, p. 52-60, Sept. 1998.

[GRA98] Gray, Paul & Watson, Hugh J. **Decision Support in the Data Warehouse**. New Jersey: Prentice Hall. 1998.

[HAGG98] Haggerty, N. **Toxic Data.** DM Review Magazine, Jun;1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet : http://www.dmreview.com/master.cfm?NavID=1986EdID=371.

[HALL99] Hall, Curt. **Exploration warehouse: techniques and products**. Data Management Strategies. Mar-1999. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.cutter.com/bia/dms99039.html.

[HILL98] Hill, Janelle - member of the Gartner Group. VSSymposium / ITXPO 98 - **DW Data Preparation** – conference presentation Out -1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet: http://www.gartner3.gartnerweb.com/glive/staticussym98-22h.html.

[IDCE99] I.D. Centric. **Data quality**. White papers. FirstLogic. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.firstlogic.com/downloads/_includes.

[INM97] Inmon, William H. **Como construir o Data Warehouse**. Rio de Janeiro: Editora Campus,1997.

[KIM98L] Kimball, R.; Reeves, L.; Thornwaite, W. **The *Data Warehouse* Lifecycle Toolkit: Tools And Techniques for Designing, Developing And Deploying *Data Marts* And *Data Warehouse*** - John Wiley & Sons, 1998.

[KIM98T] Kimball, Ralph. **Data Warehouse Toolkit**. São Paulo: Makron Books, 1998.

[KIM996] Kimball, R. **Dealing with Dirty Data.** DBMS Magazine, Set-1996. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.dbmsmag.com9609d14.html.

[KIM696] Kimball, R. **Mastering Data Extraction .** DBMS Magazine, Jun-1996. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.dbmsmag.com9606d05.html.

[KIM1198] Kimball, R. **Coping with the brave new requirements.** *Intelligent Enterprise.* Nov-1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.intelligententerprise.com/9811/*warehouse*.shtml.

[KIM1098] Kimball, R. **Brave new requirements for Data Warehousing**. Intelligent Enterprise. Out-1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.intelligententerprise.com/9810/*warehouse*.shtml.

[KOR95] Korth, Henry F. **Sistema de banco de dados.** São Paulo: Editora McGraw-Hill, 1995.

[LARR99] Larry P. English. **Data Cleansing in the *Data Warehouse.*** DM Review Magazine, Dec-1999. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.dmreview.com/master.cfm?NavID=198&EdID=1669.

[MALL99] Malloy, Amy. **Tutorial Data Mart**. Computerworld 1999. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.computerworld.com.

[MORI98] Moriarty, Terry & Hellwege, Suzi. **Criteria to help you decide if a packaged migration tool fits your organization.**Mar-1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet em http://www.inastrol.com\articles\9803.htm.

[MOSS98] Moss, L. **Data Cleasing: A Dichotomy of *Data Warehouse.*** DM Review Magazine, Fev;1998. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.dmreview.com/master.cfm?NavID=55&EdID=828.

[ORR97] Orr, Ken. **Data Warehousing Technology**. The Ken Orr Instititute. White paper. 1996-1997. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.kenorrinst.com/dwpaper.html.

[POE98] POE, Vidette, KLAUER, Patricia, BROBST, Stephen. **Building a *Data Warehouse* for decision support*.*** New Jersey: Prentice Hall PTR, 1998.

[SHEP99] John B. Shepherd. **Data Migration Strategies**. DM Review Magazine, jun-1999. Capturado em 15 de Agosto de 2000. Online. Disponível na internet. http://www.dmreview.com/master.cfm?NavID=198&EdID=996.

[SQUI95] Squire, Cass. **Data Extraction and Transformation for the Data Warehouse**. Communications of the ACM, 446-447. Jun-1995.

[SRIV99] Srivastava, Jaideep & Chen, Ping_Yao. ***Warehouse* creation: a potential roadblock to data warehousing**. IEEE transactions on knowledge and data engineering, vol.11, nr.1, Jan,Fev - 1999.

Links:

http: //www.*dwinfocenter*.org

http: //www.informatica.com

http: //www.*dw-institute*.com

http: //www.*datamation*.com

http: //www.*atre*.com

http: //www.*data-warehouse*.com