

Pesquisa de Data Warehouse
Fundamentos
ver 1.0

André Luiz Cabral Dutra
22 de janeiro de 2006
www.andrecabral.eti.br

Índice

1. O QUE É UM DATA WAREHOUSE?	3
1.1 OPERAÇÃO	3
1.1.1 <i>CARACTERÍSTICAS</i>	4
1.2 INTEGRIDADE RELACIONAL	5
2. INFRA-ESTRUTURA E ARQUITETURAS	5
2.1 ARMAZENAMENTO E EXTRAÇÃO	6
2.1.1 <i>CRIAÇÃO DA BASE</i>	7
2.1.2 <i>ATUALIZAÇÃO DA BASE</i>	8
2.1.3 <i>LIMPEZA DOS DADOS</i>	8
2.2 INTEGRAÇÃO DOS DADOS	9
2.2.1 <i>PERSONALIZANDO A BASE</i>	9
2.2.2 <i>DATA MARTS</i>	10
3. ESTRUTURA DE CONSULTAS	10
3.1 ESTRUTURA STAR	11
3.2 ESTRUTURA SNOWFLAKE	12
4. ETL (EXTRACT, TRANSFORM, LOAD)	12
4.1 EXTRACT	12
4.1.1 <i>ÁREA DE ADAPTAÇÃO</i>	13
4.2 TRANSFORM	13
4.3 LOAD	13
5. CONCLUSÕES	14
BIBLIOGRAFIA	15

1. O que é um Data Warehouse?

Um Data Warehouse é um banco de dados reunido a partir de muitos sistemas e destinados a oferecer suporte à produção de relatórios gerenciais e à tomada de decisão (inteligência empresarial). É composto de diversas bases de dados que são exportadas para ele; é necessário sua interação com a base de dados de operação da empresa para obter informações internas relacionadas ao seu funcionamento, e utilizar bases de dados de fatores externos como informações sobre clientes, custos sobre determinado setor, concorrentes ou quaisquer outros que sejam de interesse para poder ajudar e fornecer possibilidades na tomada de decisão.

A “exploração de dados” é uma das formas mais utilizadas para a criação de modelos previsíveis e as principais razões para se manter um Armazenamento de dados são:

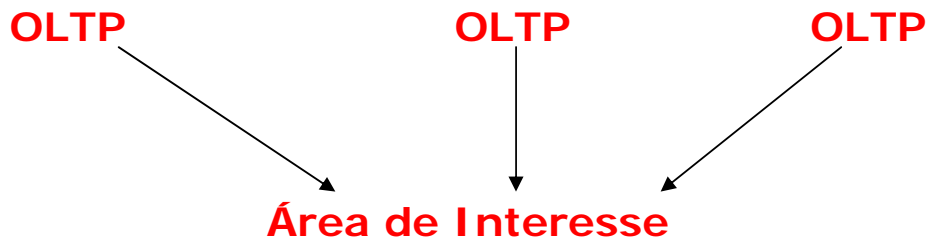
- Fornecer confiabilidade de informações corporativas;
- Proporcionar acessos a dados sem a dependência de relatórios produzidos por terceiros;
- Permitir a análise de dados corporativos e produzir fatores condicionais.

Os motivos de se implementar um Data Warehouse são para fornecer acessibilidade, prazo para levantamento de informações, formatação dos dados e a integridade das informações para possibilitar tomadas de decisões com uma base sólida.

1.1 Operação

No Data Warehouse, pode-se fazer a correlação entre os itens sem que haja um limite real, já que trabalha-se de áreas e segmentos que sejam, a princípio independentes, para quaisquer outras que possam ser de interesse para com isso, obter a análise de possíveis padrões ou correlações que possam ajudar na definição estratégica da Empresa ou mesmo sobre determinada tecnologia ou metodologia como CRM e ERP. Porém, é necessário que se tenha conhecimento sobre o que se quer analisar, pois uma pesquisa a nível macro e muito abrangente, pode sobrecarregar o sistema e causar total indisponibilidade do mesmo e gerar relatórios equivocados.

O uso de base de dados OLTP em grandes quantidades e de grandes extensões de forma indiscriminada e sem análise de nenhum tipo de padrão, não irá resultar em nenhuma informação relevante; neste caso, deveria ser utilizado um padrão orientado para assunto como venda e compra e selecionar os OLTPs associados aos produtos e serviços; ou fazer o caminho inverso, da origem dos dados.



Para melhor definir a forma que deverá ser utilizada, deve-se responder a 3 perguntas durante a elaboração do DW:

É necessário um acesso mais rápido aos dados? Quanto mais rápido?

Qual o nível de detalhe de relatório e o grau de confiabilidade desejada?

Será permitida a manipulação dos dados até que nível pelos usuários?

Quais serviços irão utilizar o DW?

Com esses dados, é possível começar a estipular um ROI e poder prever o nível de investimento ideal com base no modelo de operação atual e o crescimento almejado, já que um DW é recomendado para empresas que desejam o crescimento no setor, já que ele nada mais é do que um componente estratégico de produção de relatórios.

Quanto ao custo do desenvolvimento de metodologia de acesso as bases de dados do DW, não há restrições, visto que estas podem ser feitas com praticamente qualquer tecnologia; o que se deve dar ênfase na elaboração de um DW é para a estrutura de dados e possibilitar a maior flexibilidade possível de acesso ao banco.

1.1.1 Características

- Integralidade

Deve ser capaz de interpretar e analisar dados equivalentes em base de dados distintas, com formatos diferentes;

- Integridade

Um Data Warehouse não deve permitir que ninguém altere a base de dados; como esta é um espelho da base operacional, alterar o DW, seria como alterar a história de determinado produto ou serviço;

- Variações de tempo

Uma pesquisa tem que ser capaz de determinar a variação no tempo exata para qualquer solicitação, sem isso, os dados ficam sem uma base de sustentação.

1.2 Integridade Relacional

É um recurso de banco de dados para manter a consistência da base. O grande problema do seu uso é o decréscimo da performance do servidor, que normalmente, estes processos não são contabilizados no processo de planejamento. O não uso desta ferramenta implica na possibilidade de corromper bases de dados do DW. Existe a cultura de não implementar esta facilidade pelo fato de se estar desenvolvendo um Data Warehouse, e como ele não aceita que sejam feitas modificações em sua base, vários projetos desconsideram o seu uso, mas existe a funcionalidade do PQO (Parallel Query Option) que exige 2 processadores, mas aumenta a performance de forma significativa.

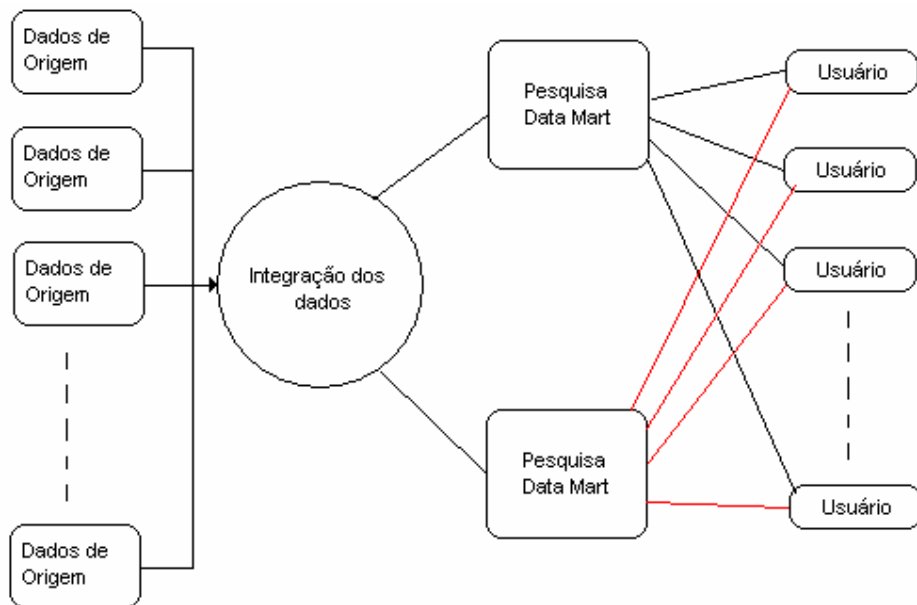
A integridade relacional pode ser efetuada por meio de regras e restrições de integridade quando não for de interesse habilitar o recurso nativo da aplicação. Um exemplo é o uso da função ENABLE NOVALIDATE. Outro ponto onde é muito útil o uso de IR fora da aplicação é durante a transformação dos dados. Ela poderá verificar se há algo errado e corrigir antes que as informações cheguem aos clientes.

O conceito de integridade relacional é que existem várias bases de dados com dados correlacionados. Um exemplo seria uma venda de um produto via WEB; é possível que o pedido fique sem um produto vinculado após a solicitação do cancelamento do produto, mas é inviável que exista um produto vendido sem um pedido; neste caso, se determinado registro relacionado a um pedido for apagado, o seu relacionamento a base de produtos também será apagada, automaticamente.

2. Infra-Estrutura e Arquiteturas

É necessário que a estrutura seja capaz de obter dados de diversas fontes; integrar os dados em um repositório comum após terem sido extraídos e limpos de suas fontes de forma a manter a integridade; formatar os dados e desenvolver aplicações para possibilitar a sua utilização por todos os usuários.

Existem várias formas de se fazer isso, ou seja, várias arquiteturas que podem ser implementadas, tudo dependendo da necessidade. Mas independente do modelo escolhido, haverá momentos em que será necessário o uso de informações muito antigas e estas, na maioria das vezes, estarão em fita e necessitam ser otimizadas, mas sem perda de conteúdo, como um resumo mensal ao invés de dados diários. O modelo mais usual seria da seguinte forma:



Onde se tem um sistema centralizado de integração de dados mas também possui os Data Marts que podem ser vistos como um DW setorizado para atender a um determinado público, o que também otimiza o processamento. Ela pode ser implementada de Top-Botton ou Botton-Top; as diferenças são que no primeiro, se tem uma escalabilidade quase infinita, já que se elabora o concentrador de dados e depois os Data Marts; já no segundo, a evolução fica presa as possibilidades que forem implementadas antes de se vincular o depósito de dados aos Data Marts existentes, ou mesmo limitando a arquitetura ao uso único e exclusivo de DMs. Esta última opção citada só válida para ambientes limitados e que não necessitem compartilhar as suas informações com o restante da corporação.

Outra arquitetura seria a centralizada, eliminando-se os Data Marts. Neste caso, tem que se saber exatamente até onde se quer levar as pesquisas e elaboração de relatórios, visto que não permite muita flexibilidade e possui muita demanda de processamento. O projeto de implementação deve ser extremamente rigoroso e é muito complexo.

Em ambos os casos, as ferramentas que fazem a interface com o usuário trabalham com OLAP, que por sua vez possibilita analisar dados atuais, muito antigos e projeções em sistema multidimensionais.

2.1 Armazenamento e Extração

Independente da arquitetura desenhada, todas tem que se preocupar na forma de como e quando irão exportar seus dados da base operacional OLTP para a base do warehouse, OLAP. Existem diversos complicadores que podem ser agravados de acordo

com a exigência do projeto, visto que os sistemas OLTP são previsíveis e projetados para uso em poucos registros simultaneamente, já o OLAP não possui uma metodologia pré-definida, e trabalha com milhares de registros simultaneamente.

Considerando uma das principais características que é a de ser um sistema não-volátil, não deve ser feita uma atualização completa da base do DW, visto que iria interromper com todo o ciclo dos dados e interromper o registro histórico; lembrando que o DW é um livro completo de história da empresa e de seus clientes e serviços.

Outro ponto contra atualizações completas é o tempo e o custo da banda ocupada para a transmissão dos dados, já que bases para um warehouse tendem a ser enormes.

2.1.1 Criação da base

A elaboração da base de um DW ou de um DM deve ser feita durante o projeto dos mesmos por meio de pesquisas com os usuários e patrocinadores para determinar quais dados são relevantes para o DW. Nunca deve-se colocar tudo no warehouse, se assim o fizer, arcará com o custo de baixíssima performance e alto custo de manutenção; e uma descrição rápida tirada do Oracle 8i Data Warehouse descreve a base do DW como sendo “... um banco de dados normalizado é como um arquivo simples que foi dividido em arquivos menores ou tabelas, para armazenar os dados mais eficientemente.” (pág. 137)

Para fazer o armazenamento dos dados selecionados no DW, os dados precisam, na sua maioria, serem convertidos e durante esse processo, começam a ser elaboradas as ferramentas que irão alimentar, de tempos em tempos, ou em tempo real o DW. Deve-se prestar bastante atenção nas informações que sejam relacionadas a comprovação da integridade dos dados e de suas origens. Sempre tenha esses dados relacionais armazenados para atualizações e pesquisas de possíveis inconsistências que venham a ocorrer.

Todo o processo irá ser totalmente dependente de como a base de dados operacional foi elaborada. Se os dados estiverem em ASCII será muito simples transportá-los para o DW. Tudo irá depender do que irá encontrar e para o que quer transformar; e para manter o sistema mais flexível e propenso a mudanças, recomenda-se a elaboração de sistemas de extração de dados das bases de origem separados dos sistemas que as carregaram para o sistema do warehouse. Com isso os sistemas também ficam isolados aumentando a tolerância a falhas são independentes, o que pode permitir um futuro upgrade de tecnologia, caso seja necessário.

2.1.2 Atualização da base

É necessário muita coerência durante a atualização da base. Existem muitas formas de elaborar um processo ou rotina para atualizar a base com fundamento na alteração dos dados de origem. A forma mais fácil de determinar isso é se na tabela de origem existem marcadores de tempo, ou seja, um registro que indique a quanto tempo determinado dado foi alterado. Com isso basta comparar este dado de tempo com o DW e determinar se é ou não necessário atualizar o registro. O problema é quando a atualização é uma exclusão, uma ação de apagar um registro em que não se guarda nenhum registro temporal na base de origem.

Outro método é através de marcadores de acionamento (nem todas as tecnologias suportam). É normalmente utilizado como técnica de segurança, já que grava todas as ações executadas em uma base de dados em um arquivo de log contendo informações sobre data e hora da alteração. Bastaria que o aplicativo tivesse a capacidade de interpretar o log e com isso determinar quais registros devem ser atualizados. Os grandes problemas são quando não se tem esses acionadores e é necessário aplicá-los e a perda de desempenho que estes irão sofrer. Como é necessário mexer na base de dados operacionais, passa a ser mais uma questão política do que técnica.

Atualização em tempo real pode ser obtida através da tecnologia AIS (Application Integration Software). É necessário que os sistemas de origem já a utilizem e baseia-se no princípio de que quando determinado dado é gerado ou alterado, ele é o responsável por enviar uma notificação a todos os clientes informando da atualização; com isso poderia adicionar o warehouse como sendo um cliente do AIS e as atualizações seriam quase em tempo real. A dificuldade deste método é que ele pode ser centrado em uma base que pode estar sendo usada para diversas razões como interface com CDNs, controles em sistemas e-commerce, sistemas B2B; e todos com focos diferentes como tradução de padrões para maior facilidade de entendimento por parte dos usuários, neste caso ele seria como um front-end em um sistema com diversas tecnologias e sistemas; poderia ser um grande depósito de dados (esta seria a melhor condição para um DW), mas ele pode estar sem a granularidade necessária para uma boa implementação de DW.

2.1.3 Limpeza dos dados

Deve-se ter em mente que para se ter um bom aproveitamento do warehouse, os dados carregados nele devem ser limpos. Informações limpas são as que possuem referência e não estão replicados em diversas bases sem uma referência cruzada; não devem possuir homônimos em nenhuma base; e que respeitem o formato do campo. Em alguns casos, eles podem estar corretos, mas não padronizados, como endereços e códigos de identificação.

A melhor forma de resolver estes problemas é nas suas origens, ou seja, no sistema OLTP de produção. A correção durante o processo de migração dos dados será muito dispendiosa e poderá causar problemas em futuras expansões de bases de origens ou da adição de novas bases. Lembre-se de que dados sujos significam um DW não confiável.

2.2 Integração dos dados

Como visto na seção de arquitetura do warehouse, a camada de integração não é obrigatória, mas a sua utilização é extremamente recomendada para possibilitar a expansão do warehouse. Aqui será mostrada a razão disso.

Imagine a situação de se obter um warehouse centralizado em diversos data marts, que são as estruturas que são realmente utilizadas para consulta. Seria necessário que cada DM tenha o seu repositório de dados em um trabalho equivalente a (limpeza de dados x DM) x quantidade de DMs. Dá para perceber que o trabalho será muito maior e a probabilidade de inconsistência de dados entre os DMs seria grande assim como a duplicação de dados, já que cada um iria tratar e armazenar a informação de uma forma e o sistema final ficaria parecendo uma rede em mesh (todos os pontos estão interconectados).

Utilizando uma camada para integrar os dados, todos os DMs terão acesso a mesma base de informações permitindo que todos os setores que os utilizem possam tomar decisões de forma apropriada e sem a possibilidade de conflitos de informações ou falta delas.

2.2.1 Personalizando a base

Existem condições que necessitam que a estrutura do DW seja totalmente diferente da de origem. A situação mais usual é o não uso dos campos chaves de origem como tendo estas mesmas características na base do DW. Costuma-se implementar um campo para chaves extra (Chaves substitutas – *Surrogate Keys*), sendo que estes não possuem nenhuma informação útil para o usuário, exceto o ato de controle e ordenação no DW e possibilitar que informações que no sistema operacional estariam sem uso por não estarem padronizadas, possam ser utilizadas, analisadas e com isso determinado se sua incidência é uma constante ou somente um fator eventual.

Outra questão é sobre auditoria, autenticidade e integridade dos dados. Com a nova lei Sarbanes-Oxley impulsionada pela onda de fraudes, faz-se necessário efetuar um controle de tempo e eventos que circundam todo o histórico de dados com informações sobre datas de modificações, registros correntes e armazenamento de registros substituídos; isto requer o uso de alguns campos que não são normalmente acessíveis

pelos usuários e um certo aumento da administração da base.

2.2.2 Data Marts

Os Data Marts irão necessitar de um carregamento de dados da base integrada, ou em uma arquitetura isolada, eles estariam diretamente vinculados ao sistema OLTP, sendo que este teria que sofrer algumas alterações e dificilmente permitira a correlação entre bases distintas.

Por meio do uso de uma base integrada, possui a grande vantagem de que toda a informação já foi tratada é só precisa ser movida, permitindo uma agilidade muito maior para a elaboração dos DMs. Podem vir a existir alguns problemas como alterações da base integrada que não foi repassada para a base dos DMs após alguma alteração de registros, mas isto é resolvido facilmente ou pela atualização dos dados de imediato ou de forma manual pelo usuário no relatório final.

A base dos DMs também pode ser comprimida para fatores semanais, mensais, anuais ou qualquer outra que se deseje. Com isso, é mantido o nível de detalhe no DW, sendo que este é resumido no DM, caso seja necessário mais detalhes ou outras informações, bastaria acessar o DW e obtê-las; como este é um processo que não será comum, pelo menos não deveria, o ganho de desempenho é enorme. Seu sistema de gerenciamento pode ser baseado em métodos relacionais ou multidimensionais.

3. Estrutura de consultas

Existem alguns modelos de estruturas para possibilitar que os usuários possam realizar suas consultas da forma mais otimizada possível. Mas a facilidade para os usuários significa uma necessidade de trabalho administrativo maior da base de dados, ainda mais devido a que as consultas possuem uma necessidade de informações desnormalizadas com diversas origens.

Por esta razão, se a empresa usuária tiver a expectativa de uma ampliação gradativa do warehouse, deve optar pela elaboração de projetos top-down que possibilitarão maior flexibilidade e controle sobre que tipo de informação estará disponibilizada nos Data Marts para os usuários usufruírem da melhor forma possível e possibilitar mudanças na estrutura de elaboração de relatórios, seja pelo ato de acrescentar mais uma informação ou retirar. Imaginem como seria dispendioso e trabalhoso fazer isso se toda a estrutura estivesse no DM e fosse necessário importar um campo de uma tabela OLTP? Imaginem a quantidade de mudanças e a interrupção do serviço que seriam geradas.

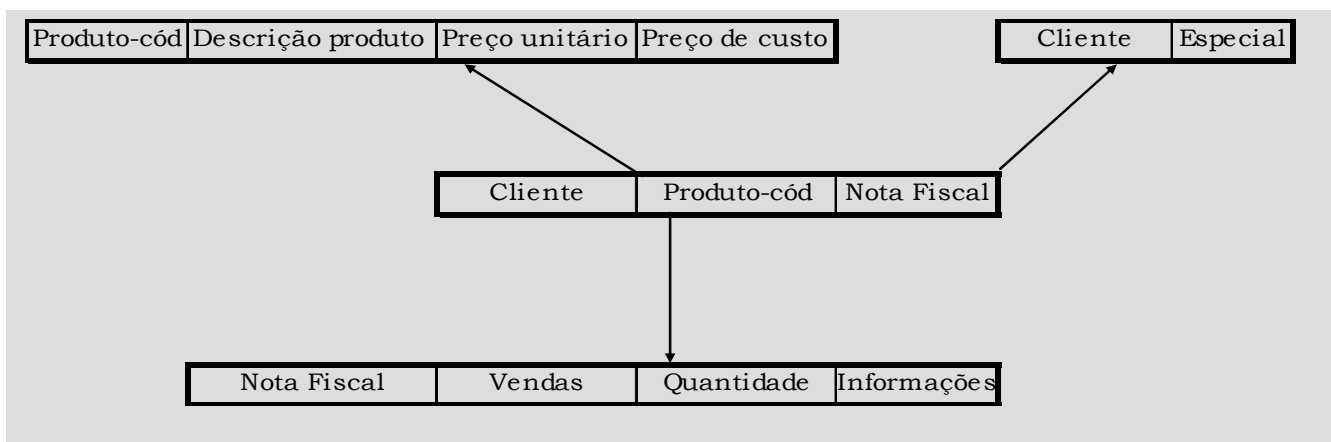
3.1 Estrutura Star

É a forma mais comum de se encontrar um DM. Consiste em diversas tabelas correlacionadas entre si. Isoladamente, parece uma “coisa”, uma teia de aranha sem nexos algum, mas prestando atenção a sua estrutura, verificasse que todas as tabelas possuem um vínculo com a tabela central. Na verdade, este vínculo é uma correlação de chaves de forma a possibilitar diversas referências cruzadas e diminuindo, consideravelmente, o tamanho de disco e o processamento necessário, aumentando o desempenho significativamente.

O grande problema é a alta complexibilidade deste tipo de base que necessita de um controle muito específico e de verificações de integridade relacional para novos itens inseridos na base.

Sua estrutura é feita de forma que a tabela central, considerada de tabela de fatos, contenha todos os dados chaves que serão utilizados; e as outras tabelas, que são chamadas de dimensionais, são criadas a partir da tabela de fatos, possuindo uma estrutura desnormalizada, como já fora dito antes, já que provavelmente irá conter colunas de diversas origens OLTP. A tabela de fatos deve representar as principais informações da empresa para o ponto de análise e elaboração de relatórios solicitado.

Também se faz necessária a criação de campos que contenham informações referentes à origem dos dados, isso se faz necessário por questões de auditoria e manutenção, já que é uma informação praticamente inútil para os usuários. Um exemplo de sua estrutura é o que segue abaixo:



Pode-se verificar que a tabela de fatos (central) possui itens que, isoladamente, pertenceriam a cada uma, ou várias das outras tabelas. Com isso já dá para imaginar o espaço em disco que se é economizado e o aumento de desempenho. Agora imaginem todos esses dados misturados em uma única tabela? Como se poderia fazer uma

pesquisa sobre quais produtos venderam mais ou menos? Que cliente comprou mais? Qual está sendo o lucro real?

3.2 Estrutura Snowflake

No esquema Snowflake, existem algumas diferenças substanciais em comparação ao modelo Star. O princípio é o mesmo, só que as tabelas de dimensões são normalizadas e de tamanho padrão entre si. Outro ponto é que elas não são mais relacionadas diretamente a tabela de fatos; irão existir tabelas de dimensões que serão as chaves de outras tabelas dimensionais.

Os primeiros pontos verificados, são contra este modelo, já que irá demandar maior tempo de processamento, visto que a tabela com os dados terá que ser localizada em diversas análises; outro ponto é que com a normalização das tabelas de dimensão, perde-se parte da granularidade das informações para manter uma base pequena, e para mantê-la, é necessário criar muitas outras tabelas pequenas, aumento em muito as interligações entre elas e conseqüentemente, reduzindo o tempo de resposta para o usuário final.

4. ETL (Extract, Transform, Load)

O processo de carregar os dados para um Data Warehouse ou Data Mart e disponibiliza-lo para uso dos clientes é chamado de ETL; onde *EXTRACT* é a função de extrair os dados das bases de origens para as de destino; *TRANSFORM* tem a função de adaptar os dados. No caso do DW, faz a sua normalização; e *LOAD* disponibiliza às informações para os clientes já que atualiza às tabelas de destino.

Nas três etapas do processo, podem ser implementadas ferramentas proprietárias ou efetuar o desenvolvimento internamente de alternativas. A escolha irá depender de como está o ambiente de origem, das informações coletadas e qual é o ambiente de destino; mas na grande maioria dos casos, a melhor solução é misturar ambas, já que uma solução proprietária provavelmente não terá como se adaptar perfeitamente a estrutura operacional do cliente.

4.1 Extract

Para que a extração de dados ocorra de forma correta e eficiente, é necessário que a aplicação que estiver executando a tarefa, tenha total compatibilidade para com a arquitetura de dados da origem. Para isso, é necessário implementar algum nível de

código nos sistemas de origem para possibilitar o que é chamado de CDC (Change Data Capture). Isso possibilitará que a manutenção do relacionamento de tempo entre registros de bases de dados de origens diferentes e a do warehouse sejam mantidas, mantendo assim, a consistência temporal dos fatos e sua integridade histórica; permite também, que sistemas de log de eventos possam indicar quais campos não foram corretamente extraídos e quais sofreram alterações. Este relacionamento é mais bem realizado se a chave primária da tabela for usada como índice, com isso pode-se economizar o espaço em disco necessário para o log.

A ação de acessar os dados e copiá-los para seu destino pode ser feita, praticamente, de todas as formas de conectividade existentes; tudo vai depender das limitações do sistema de origem, se houver alguma nesse sentido. É conveniente abster esse ponto durante o desenvolvimento do ETL e considerar todas as fontes de origem como sendo uma coisa só. Isso permitirá uma visão lógica global do projeto.

4.1.1 Área de Adaptação

É um espaço que pode ser virtual ou físico. Seu objetivo é possibilitar que os dados sejam extraídos e carregados nele para que sejam modelados, ou seja, adaptados para seus repositórios finais (DWs e DMs); também garantem que os sistemas finais não sofram interrupções para possibilitar o carregamento de informações.

O uso de soluções que trabalham com essa tecnologia aumenta muito o custo final de um DW, mas permite que seu funcionamento seja praticamente ininterrupto e permite fazer a transformação de uma só vez e ainda fornece um certo grau de restore em caso de desastre, mas não full restore.

4.2 Transform

Normalmente a transformação ocorre durante ou após a extração, mas não existe uma norma, sendo que a transformação, que nada mais é do que adaptar os dados para exibirem as informações solicitadas, pode ocorrer até durante a consulta, mas para isso, deve-se verificar a necessidade e possibilidade de cada passo.

Quando a quantidade de dados é muito grande, pode ser interessante distribuir o processo de transformação entre as 3 etapas do ETL; quando houverem dados que não são acessados com frequência, eles podem ser transformados quando solicitados.

4.3 Load

Funciona por meio de comparações de conteúdo dos registros que tem a mesma chave primária, mas que possuem algum dado diferente entre si (uma atualização de endereço

por exemplo), neste caso será criado um novo registro para acomodar a alteração e o antigo não será apagado e sim “desativado”; outro processo é o de verificar alguma chave primária que exista nas tabelas de origem, mas que não estejam relacionadas nas tabelas do DWs e/ou DMs. Neste caso, o registro inteiro, conforme foi transformado, será inserido na tabela.

Esses processos são válidos para as tabelas dinâmicas. Já para as tabelas de fatos, que consistem de fatos que são considerados o coração do cliente, às atualizações só podem ocorrer quando os novos campos tiverem suas ponteiros apontados corretamente para as tabelas de dimensões necessárias.

5. Conclusões

Um Data Warehouse fornece uma possibilidade enorme para aumentar os horizontes de visões estratégicas. Existem diversas formas de implementá-lo com praticamente todos os tipos de tecnologias e linguagens possíveis distribuídas entre todas as suas etapas.

Para escolher qual a melhor forma de implementar um DW, é uma necessidade básica que se conheça, profundamente, a estrutura do negócio do cliente, suas necessidades e anseios; e nunca acreditar que o que haverá no DW será uma cópia do sistema de origem.

Bibliografia

Oracle 8i Data Warehouse, Editora: Oracle Press

<http://www.idc.com/>

<http://genesis.nce.ufrj.br/dataware/>

<http://labc.icmc.usp.br/didatico/dw/>

<http://www.datawarehouse.inf.br/>

<http://www.liacc.up.pt/~jgama/Bdc/olap.pdf>

http://www.eacnet.com.br/integridade_relacional

http://www.fw.uri.br/~adrovane/bd1_arquivos/integridade.pdf