

Criando *data warehouse* com o modelo dimensional

Donizete Carlos Bruzarosco*, André Vinicius Castoldi e Roberto Carlos dos Santos Pacheco

¹Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 88037-001, Florianópolis-Santa Catarina, Brazil. *Author for correspondence. e-mail: dcbruzar@eps.ufsc.br

RESUMO. O conhecimento armazenado ao longo dos anos nas bases de dados organizacionais é um bem vital para a sobrevivência de uma empresa neste mercado competitivo. Para a sua efetiva utilização como apoio ao processo decisório de uma organização, os dados precisam estar organizados de uma forma que possibilite o seu acesso rápido e fácil entendimento. Este trabalho apresenta a modelagem dimensional como alternativa para a construção de um *data warehouse*, mostrando os seus elementos, algumas técnicas de modelagem, assim como os passos para o seu desenvolvimento, atendendo assim aos requisitos citados. Também apresenta os resultados e considerações, de um estudo de caso, com a aplicação do modelo dimensional para a criação de um *data mart*, para um portal jurídico da internet

Palavras-chave: armazém de dados, modelagem dimensional, suporte a decisão.

ABSTRACT. Developing *data warehouse* using dimensional model. Stored knowledge in organizational databases during the years is vital for the survival of a company in a competitive business world. For its effective use as support in a decision-making process of a organization, data must be organized for fast access and easy comprehension. A dimensional modeling is provide as an alternative for the construction of a *data warehouse*, with its elements, modeling techniques, and steps for its development to meet the mentioned requirements. Results and considerations about a case study are given, together with the application of a dimensional model for the creation of a *data mart*, for a juridical internet portal.

Key words: *data warehouse*, dimensional modeling, support in decision-making.

A globalização tornou a competição acirrada e o conhecimento fundamental para que as organizações possam vencer nesse mercado competitivo. Para se obter esse conhecimento, que é vital para apoiar as empresas nas tomadas de decisões estratégicas e gerenciais, é necessário estruturar os dados de forma que facilite a sua compreensão e extração de forma eficiente. Os dados de uma organização podem ser armazenados de duas formas: por meio do registro dos sistemas de informações operacionais e *data warehouse* (Kimball, 1998).

A modelagem dimensional atende a dois requisitos fundamentais para o sucesso de um *data warehouse*: simplicidade na organização dos dados, permitindo seu fácil entendimento por usuários finais; e um bom desempenho na geração de consultas e relatórios de apoio aos executivos nos processos decisórios, pela redução significativa de operações de junções de dados.

O enfoque deste trabalho é a aplicação do modelo dimensional no desenvolvimento de um *data warehouse*, atendendo à fase de modelagem, a qual é uma das fases que compõe o ciclo de vida de projeto de um *data warehouse*¹ e, portanto, as demais fases não são mencionadas.

Este trabalho apresenta inicialmente o que é um *data warehouse*, como planejá-lo e por que o modelo dimensional é o mais adequado do que o modelo relacional, no seu desenvolvimento. As seções seguintes definem os elementos, algumas técnicas e os passos envolvidos na construção de uma modelagem dimensional de um *data warehouse*. Na seqüência são mostrados os resultados de um estudo de caso da criação de um *data mart* para um portal

¹ Fases do ciclo de vida do projeto de um *data warehouse* (Kimball, 1998): planejamento do projeto, definição dos requisitos do negócio, projeto da arquitetura técnica, seleção e instalação de produtos, modelagem dimensional, projeto físico, desenvolvimento do projeto e *data staging*, especificação da aplicação do usuário final, implantação e manutenção.

jurídico, utilizando o modelo dimensional. Por fim, são apresentadas as conclusões deste trabalho.

Data warehouse

A força do *data warehouse* dentro das empresas está na organização e na entrega de dados que apóiem o processo de tomada de decisão. O *data warehouse* é a implementação física de um modelo de apoio a decisões, que disponibiliza à empresa as informações necessárias à tomada de decisões estratégicas (Inmon, 1995). Um *data warehouse* é um repositório de dados com as seguintes características (Inmon, 1995):

- **baseado em assuntos:** refere-se ao fato de que o *data warehouse* está organizado de maneira que descreve o **desempenho** dos negócios; os bancos de dados operacionais são orientados para o **processo** dos negócios;
- **integrados:** refere-se ao fato de os dados serem organizados para fornecer uma fonte única;
- **variáveis em relação ao tempo:** reconhecem que o desempenho de negócio é medido em pontos cronológicos (final do mês, por exemplo) e comparado em relação ao tempo;
- **não-voláteis:** sugere que os dados, uma vez inseridos no *data warehouse*, não devem mudar; o conteúdo dos bancos de dados operacionais muda cada vez que uma transação é processada.

Um *data warehouse* suporta análise de negócios e tomada de decisões através da **integração de dados** procedentes de diversos sistemas - muitas vezes incompatíveis entre si - em uma base de dados consolidada. Esta transformação de dados em informações significativas capacita os executivos de uma organização a efetuar análises muito mais reais, precisas e consistentes (Meyer, 1998). Em outras palavras, o *data warehouse* é uma base de dados projetada especificamente para atender às necessidades de **tomada de decisão**, em vez de atender aos sistemas de processamento de transações (Daphne, 1999). Pode também ser usado como suporte a sistemas de *data mining*² (Freitas, 1998).

Os projetos lógico e físico são considerados a base do *data warehouse*. A abordagem de modelagem dimensional, utilizada para o desenvolvimento do projeto lógico, é a melhor maneira de modelar dados de suporte à decisão e oferece os melhores

resultados em termos de facilidade de uso e alto desempenho (Kimball, 1998).

Planejamento da construção de um *data warehouse*

Uma das maiores preocupações em *data warehouse* é como planejar a sua construção. Ele pode ser construído inteiro, abrangendo toda a organização de uma só vez (abordagem monolítica), ou então por áreas (Kimball, 1998). A maioria dos técnicos em *data warehouse* utilizam alguma espécie de abordagem por etapas para construir um *data warehouse* de uma empresa inteira. Em cada etapa é construído um *data mart*, que é um *data warehouse* direcionado a apenas uma área ou assunto específico da organização. Os planejadores de *data warehouse* utilizam a estratégia de construí-lo em pedaços pequenos (*data mart*) por vez, até chegar à sua totalidade. O grande problema apresentado por essa construção incremental é o risco de esses *data marts* apresentarem incompatibilidades de integração para formar o todo (*data warehouse*).

A solução para essa abordagem evolutiva é iniciar o planejamento do *data warehouse* com uma fase de especificação de uma arquitetura de dados geral com metas específicas e finitas, a qual guiará a implementação dos *data marts* em separado, permitindo uma aderência entre eles (Kimball, 1998). O projeto lógico de um *data warehouse* pode ser realizado tanto através da modelagem entidade-relacionamento quanto pela modelagem dimensional (Kimball, 1998). As vantagens e desvantagens de cada abordagem são descritas a seguir.

Modelagem entidade-relacionamento. A modelagem entidade-relacionamento é uma técnica de projeto lógico de banco de dados largamente utilizada, que procura eliminar redundância e inconsistência dos dados, além de facilitar a sua manutenção. Como resultado, gera muitas tabelas, e isso tem um preço, pois os sistemas de software tornam-se complexos e ineficientes na manipulação e recuperação dos dados, porque requerem uma cuidadosa atenção no processamento de algoritmos para ligar esses conjuntos de tabelas (Kimball, 1998).

Mesmo em aplicações simples o Modelo ER desenvolve muitas tabelas, que são ligadas por uma intrincada rede de relacionamentos. Dependendo do porte da empresa, a modelagem de seus dados resulta em centenas ou milhares de entidades, causando uma série de problemas, tais como: usuários finais que não entendem ou não se lembram do diagrama de entidade-relacionamento (DER); a interface gráfica de um diagrama geral não

² *Data mining* é um processo não trivial de procura por padrões válidos, novos, potencialmente úteis e inteligíveis a partir de uma coleção de dados (Freitas, 1998).

é amigável para os usuários finais; *software* de consulta geral sobre o modelo paga um alto preço de desempenho; e a sua aplicação para *data warehouse* contraria a finalidade deste, de ser intuitivo e oferecer um alto desempenho na recuperação de dados (Kimball, 1998).

Modelagem dimensional. Modelagem dimensional é a técnica de projeto lógico de banco de dados mais usada no desenvolvimento de *data warehouses*, embora também possa ser aplicada ao projeto de sistemas de informações operacionais. Na verdade, ela busca apresentar os dados em um formato que seja intuitivo e ao mesmo atenda a acessos com um alto desempenho (Kimball, 1998). Um modelo dimensional é composto por uma tabela com uma chave composta, denominada tabela de fatos, e um conjunto de tabelas menores conhecidas como tabelas de dimensão, que possuem chaves simples (formadas por uma única coluna). Na verdade, a chave da tabela de fatos é uma combinação das chaves das tabelas de dimensão. Isto faz com que a representação gráfica do modelo dimensional assemelhe-se a uma estrela. Por este motivo, o modelo também é conhecido como modelo estrela (Kimball, 1998).

A idéia fundamental da modelagem dimensional baseia-se no fato de que quase todo tipo de dado do negócio pode ser representado como uma espécie de cubo de dados, onde as células do cubo contêm os valores medidos e os lados do cubo definem as dimensões naturais dos dados (Harrison, 1998). A Figura 1 contém um exemplo de um cubo com 3 dimensões: produto (conjunto de produtos comercializados pela empresa), mercado (área de atuação) e tempo (período de tempo de atuação). Modelos dimensionais reais do mundo dos negócios geralmente possuem entre 4 e 15 dimensões.

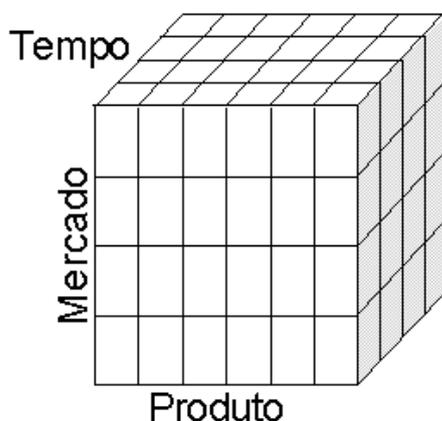


Figura 1. Exemplo de um cubo dimensional

Elementos de um modelo dimensional

A seguir são definidos os elementos que compõem o modelo dimensional (Kimball, 1998).

Dimensões. Uma dimensão é uma coleção de atributos textuais que são altamente correlacionados entre si. Os atributos textuais que descrevem coisas são organizados dentro das dimensões. Em uma base de dados de varejo são comuns dimensões como produto, armazém, cliente, promoção e tempo. A Figura 2 apresenta exemplos de dimensões: mercado, produto e tempo, de um histórico de negócios. Uma dimensão é dita conformada quando pode ser compartilhada por duas ou mais tabelas de fatos. Um exemplo de dimensões conformadas seria um caso que contivesse duas tabelas de fatos: uma de vendas realizadas e outra de vendas previstas, e que compartilhasse as mesmas dimensões: produto, mercado e de tempo.

Tabela de fatos. A tabela de fatos sintetiza o relacionamento existente entre as diversas dimensões. Isto ocorre porque a chave da tabela de fatos é a associação das chaves primárias das tabelas de dimensões. Geralmente este tipo de tabela possui um ou mais fatos numéricos que ocorrem na combinação de chaves que define cada registro. A tabela Histórico de Negócios, ilustrada na Figura 2, é um exemplo de tabelas de fatos.

Atributos. Geralmente são campos de textos, os quais descrevem uma característica de uma coisa tangível. Os atributos mais comuns são as descrições de produtos. O sabor é um exemplo de atributo do produto. Não se mede o atributo sabor de um produto. Se for necessário criar um produto com um novo sabor, cria-se um novo registro de produto.

Atributos de dimensões oferecem o conteúdo da maioria das respostas solicitadas pelos usuários. Pode-se dizer que a qualidade do *data warehouse* é medida pela qualidade dos atributos das dimensões. Uma tabela de dimensão contém muitos campos-textos descrevendo os membros de uma dimensão particular. A meta do *data warehouse* é criar atributos de tabelas de dimensão com as seguintes características: eloqüente, descritivo, completo, qualidade garantida, indexado, disponível e documentado. Exemplos de atributos são apresentados pela Figura 2, tais como: descrição de produto, produtor, marca, cor, tamanho e nível, da tabela de dimensão de produtos.

Fatos. Um modelo dimensional faz distinção entre fatos e atributos. Um atributo é usualmente alguma coisa que é conhecida com antecedência. Um fato é uma observação do mercado. Muitos fatos no mundo dos negócios são numéricos, embora alguns possam conter texto. Algumas vezes, um valor numérico como “preço padrão” parece ser um atributo da dimensão produto, pois é uma constante conhecida antecipadamente. Mas, em um exame mais cuidadoso, verifica-se que o atributo preço-padrão sofre alteração durante o ano, o que leva a alterá-lo na fase de projeto para um fato. Uma típica e correta decisão a ser tomada é considerar quase todos os campos numéricos de pontos flutuantes como fatos. Exemplos de fatos tais como: reais, unidades e preço (custo), da tabela histórico de negócios, podem ser verificados na Figura 2.

Fatos aditivos, semi-aditivos e não aditivos

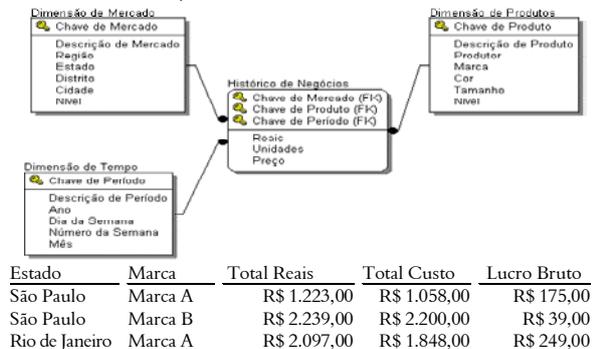


Figura 2. Exemplo de modelo dimensional e um relatório gerado a partir dele

Os fatos da tabela de fatos deveriam ser aditivos, isto é, ser a soma dos fatos ao longo de todas as dimensões (ex.: registrador de vendas). Medidas numéricas de intensidade não são perfeitamente aditivas (semi-aditivo), como por exemplo, saldo de contas e níveis de inventário, os quais são como uma fotografia de algum momento no tempo. Também é possível um fato medido ser textual (não aditivo), como por exemplo, a condição do tempo descrito pelo policial no cenário de um acidente, considerando-se que o acidente é a granularidade (o nível atômico do dado) da tabela de fatos.

A Figura 2 mostra uma modelagem dimensional que permite gerar consultas e/ou relatórios de vendas dos fatos: reais, unidades, preço (custo); por mercado (estado), produto (marca) e período (dimensões), como ilustrado no relatório criado. Os atributos “mercado”, “produto” e “período” compõem a chave primária, e também são definidos como chaves estrangeiras (*foreign keys - FK*), da tabela de fatos “Histórico de Negócios”.

A notação gráfica dos relacionamentos entre as tabelas dos exemplos das Figuras 2, 3, 4, 5 e 6 e suas respectivas cardinalidades é da linguagem e método IDEF1X (Nist, 1993), (Kern, 1999), que é utilizada para modelagem da informação baseada no modelo Entidade-Relacionamento (Chen, 1990).

Técnicas de modelagem dimensional

Algumas técnicas de modelagem dimensional são apresentadas a seguir (Kimball, 1998).

Snowflaking. Uma dimensão é dita estar *snowflaked* quando os campos de baixa cardinalidade de uma dimensão compuseram uma outra tabela, a qual foi ligada com a tabela original com chaves artificiais. O *snowflake* (ilustrado na Figura 3 pelas tabelas dimensão cliente e subdimensão região) é feito para salvar espaço em disco, mas não é recomendado para ambientes de *data warehouse* por tornar a sua apresentação mais complexa (por aumentar o número de tabelas) e para consultas com processamento mais pesado (por aumentar a necessidade de junção de tabelas para se obter a informação), acarretando maior tempo de resposta.

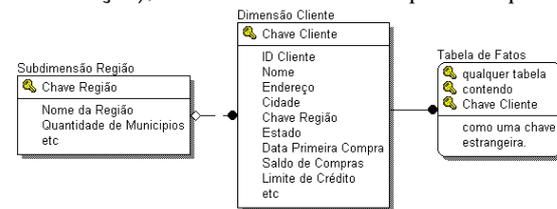


Figura 3. Exemplo de dimensão *snowflaked*

Agregação. Agregados são sumários armazenados construídos primeiramente para melhorar o desempenho de consultas. Agregados, preferencialmente são armazenados em tabelas separadas, diferentes das tabelas de fatos originais contendo os dados não agregados. Certos fatos em um ambiente podem não ser expressáveis no menor nível de granularidade. Neste caso uma tabela de agregados é requerida para apresentar esses novos fatos agregados.

Alterações em dimensões. Quando houver necessidade de se alterar uma informação, pode-se utilizar uma das três estratégias abaixo:

- **sobrepor o registro de dimensão** com os novos valores e perdendo o histórico;
- **criar um registro de dimensão adicional**, usando um valor novo da chave artificial;
- **criar um campo “velho” no registro dimensão**, para armazenar o valor prévio do atributo.

Dimensões grandes e com grande volume de alterações podem ocasionar muito consumo de espaço de armazenamento. Uma solução para esse problema é dividir a dimensão, como no caso da dimensão cliente conforme Figura 4, que para salvar espaço de armazenamento, foi dividida em duas dimensões: uma com atributos com baixo volume de alterações (dimensão cliente) e outra com atributos com alto volume de alterações (dimensão demografia). Essa solução economiza espaço em disco em detrimento do desempenho de consultas.

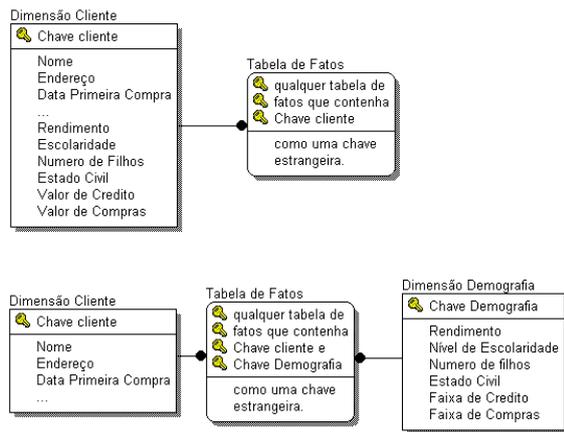


Figura 4. Divisão da dimensão cliente para suportar alterações

Chaves estrangeiras, primárias e artificiais.

Todas as tabelas dimensionais possuem chaves primárias, as quais devem ser chaves artificiais. Não se deve usar a chave original dos sistemas de produção para evitar que problemas ou alterações nos mesmos afetem o data warehouse. A chave artificial é pequena e sua única finalidade é recuperar o registro para ver seu conteúdo.

Dimensões muitos-para-muitos. Nas situações de projetos clássicos, a granularidade da tabela de fatos é facilmente entendida e é óbvio que cada dimensão esteja ligada a um registro da tabela de fatos. Frequentemente uma dimensão pode ter zero, um ou muitos valores para um dado registro. A dimensão muitos-para-muitos ocorre quando um dado registro da tabela de fatos pode ter mais de uma ocorrência em uma dimensão, sendo a solução para essa situação a criação de uma *bridge table*³.

No exemplo apresentado pela Figura 5, onde uma ocorrência da tabela de fatos **despesas do paciente** pode estar relacionada a mais de uma

ocorrência da tabela de dimensão **diagnóstico** (o paciente pode ter mais de um diagnóstico), caracterizando o relacionamento muitos-para-muitos, a solução para o problema é a criação da tabela *bridge table* **grupo de diagnósticos**.

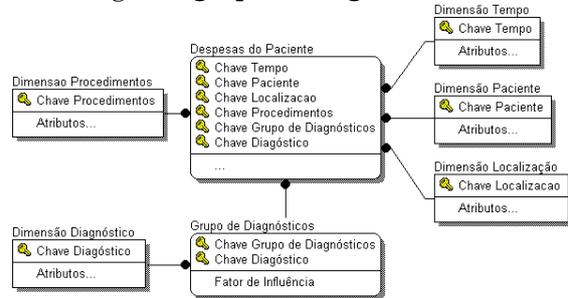


Figura 5. Exemplo de dimensão muitos-para-muitos

Hierarquia.

Representar uma estrutura de organização arbitrária é uma tarefa difícil em um ambiente relacional. Por exemplo: pode-se desejar um relatório de receitas de um conjunto de clientes comerciais relacionados entre si. Podem-se vender produtos ou serviços para quaisquer desses clientes comerciais e, em diferentes momentos, ter uma posição, individualmente ou em famílias organizadas dentro de uma estrutura hierárquica. Essa estrutura hierárquica pode ser representada com ponteiros recursivos em cada registro da dimensão cliente, contendo a chave cliente do pai na organização. Embora seja uma maneira compacta e efetiva de representar uma hierarquia arbitrária, este tipo de estrutura não pode ser usado efetivamente pela linguagem SQL. A solução para esse problema seria a construção de uma *bridge table* entre a dimensão cliente e a tabela de fatos, conforme ilustrado na Figura 6.

Drill down e *drill up* significam apresentar uma informação com mais detalhe (*drill down*) ou mais resumida (*drill up*), a partir de estruturas hierárquicas. Como exemplo de *drill down* pode-se citar uma consulta que permita mostrar em um primeiro momento os produtos vendidos por categoria, depois subcategoria, em seguida por marca e finalmente no detalhe de produto.



Figura 6. Usando uma *bridge table* para facilitar a navegação no modelo

³ Uma *bridge table* é uma tabela criada para intermediar (fazer uma ponte) em um relacionamento muito-para-muitos, entre uma tabela de fatos e uma tabela de dimensões, para resolver problemas de consultas de sistemas relacionais.

Construção do modelo dimensional

O *data mart* é construído reunindo-se um conjunto de tabelas de fatos, cada qual contendo uma chave composta pelas chaves primárias de suas dimensões (junção das chaves estrangeiras). As tabelas de fatos contêm zero ou mais fatos que representam medidas tomadas de cada combinação dos seus componentes chaves. Cada tabela de fatos é circundada por uma auréola de tabelas de dimensões e este projeto também é chamado de *esquema estrela* (Kimball, 1998), (Teorey, 1999).

A maior responsabilidade da equipe de projeto de *data warehouse* central é estabelecer, divulgar, manter e forçar o uso das dimensões conformadas. Sem uma rígida aderência proporcionada pelas dimensões conformadas, o *data warehouse* não pode funcionar como uma unidade integrada. A maioria das dimensões conformadas deve ser definida no seu nível mais granular (atômico) possível (Kimball, 1998).

Para decidir quais modelos dimensionais construir, pode-se começar com uma abordagem de planejamento *top-down* chamada matriz de arquitetura *bus* de um *data warehouse*. Essa matriz força a identificar todos os *data marts* possíveis e todas as dimensões envolvidas com esses *data marts*. Depois, pode-se ser mais específico dentro do projeto criando-se as tabelas de fatos para os *data marts*, utilizando-se do método de quatro passos (Kimball, 1998) descrito abaixo para a criação do projeto lógico:

Escolher o *data mart*. Primeiramente deve-se decidir se o *data mart* inicial tratará uma fonte simples ou múltiplas fontes. Uma fonte simples representa um domínio de aplicação de menor complexidade em relação a um de múltiplas fontes. Como exemplos de fonte simples pode-se citar: solicitação de compra, vendas, pagamentos, etc; e como exemplo de múltiplas fontes, o lucro por cliente, onde fontes que descrevem receitas devem ser combinadas com fontes que descrevem custos, para se obter o resultado desejado. É recomendado implementar separados *data marts* no contexto de um conjunto de dimensões conformadas, ligando os *data marts* em um *data warehouse bus*⁴. Comece-se com um *data mart* de fonte simples.

Declarar a granularidade da tabela de fatos. É fundamental definir claramente a granularidade do registro da tabela de fatos no projeto dimensional

proposto. Geralmente, a granularidade da tabela de fatos é escolhida para ter o menor nível de granularidade possível, pois isto permitirá responder melhor a novas consultas e a introdução de novos elementos de dados com nível maior de granularidade. Exemplos de granularidades de tabela de fato: transação de venda, indenização de seguro, transação de um caixa automático de banco, total de vendas diárias por produto, posição mensal de conta, cada item de um pedido, etc. Uma granularidade seria definida mais alta somente por razões de desempenho.

Escolher as dimensões. Após a definição clara da granularidade da tabela de fatos, a escolha das dimensões é bastante direta. A tabela de fatos em um modelo dimensional é um conjunto simultâneo de medidas de uma granularidade particular. Neste ponto, o projetista dimensional “decora” o conjunto de medidas com todas as dimensões que ele encontrar dirigidas aos requisitos do negócio. Por exemplo, o conjunto mínimo de dimensões para um item de um pedido seria data do pedido, cliente, produto e uma dimensão degenerada (não existindo fisicamente por não possuir atributos) contendo somente o número da ordem. Cada dimensão tem sua própria granularidade, a qual não pode ser menor que a granularidade da tabela de fatos

Escolher os fatos. A granularidade da tabela de fatos também permite a escolha dos fatos individuais, deixando claro qual é o seu escopo.

Um *data mart* é um conjunto coordenado de tabelas de fatos com estruturas similares (Kimball, 1998). Muitos negócios possuem um fluxo lógico que tem um começo e um fim. Um pedido, um produto ou um cliente envolvem uma série de passos. Geralmente capturam-se transações ou instantâneos fotográficos de cada passo, fazendo mais sentido gerar uma tabela de fatos para cada passo. Uma tabela de fatos e um conjunto de dimensões associadas são então definidos para cada passo da cadeia. As dimensões conformadas são cuidadosamente projetadas, e estas dimensões são usadas como o barramento (a forma de integração) para separadamente implementar cada passo na cadeia. Em um exemplo sobre seguro, talvez os dois passos mais interessantes da sua cadeia de valores sejam vendas do corretor e processamento de reclamações. Esses dois modelos dimensionais poderiam ser implementados antes dos outros com a garantia de que um *framework* de dimensões conformadas sempre guiaria os esforços de

⁴ *Data warehouse bus*: *data warehouse* composto pelos *data marts* desenvolvidos conforme uma matriz de arquitetura *bus*.

implementação subsequentes e garantiria que todas as tabelas de fatos trabalhariam juntas no futuro.

Virtualmente, cada *data mart* necessita de duas versões dos dados modelados separadamente: a versão transação e uma versão de instantâneo fotográfico periódico. Transações dão a possibilidade da mais completa visão do comportamento detalhado e o instantâneo fotográfico permite-nos rapidamente medir o *status* da empresa. Quando usados juntos, fornecem uma completa e imediata visão do negócio e, quando são parte de um *data warehouse*, eles se misturam elegantemente, formando uma visão maior através do tempo e através de outras dimensões principais.

Alguns conselhos úteis de projeto (Kimball, 1998) são:

- os nomes que são usados para identificar os *data marts*, dimensões, atributos e fatos deveriam originar-se do vocabulário do negócio;
- um atributo pode existir em somente uma dimensão, mas um fato pode ser repetido em múltiplas tabelas de fatos;
- se uma dimensão, como uma entidade do negócio parece residir em mais de um lugar, provavelmente vários papéis estão sendo representados. Neste caso, deve-se identificar unicamente os papéis e tratá-los em dimensões separadas;
- um campo simples de uma base de dados fonte pode ter uma ou mais colunas lógicas associadas com ele. Por exemplo: o campo produto pode ser traduzido para código do produto, descrição curta do produto e descrição longa do produto.
- cada fato deveria ter uma regra de agregação *default* (soma, mínimo, máximo, último, semi-aditivo, algoritmo especial e não agregável), a qual serviria para uso das consultas e relatórios e como requisito na avaliação de ferramentas geradoras de consultas e relatórios.

Os diagramas do projeto devem conter informações descritivas fornecendo um documento compreensivo do projeto, o qual também deveria conter uma breve introdução da modelagem dimensional com seus conceitos e terminologia.

Como qualquer modelagem de dados, o desenvolvimento do modelo dimensional é um processo iterativo e seu sucesso depende de um trabalho em equipe. A partir do esboço inicial do projeto, sessões de revisão com as pessoas envolvidas devem ser efetuadas, em que as informações são apresentadas e discussões são encorajadas para

explorar o suporte do modelo ao negócio. No final do projeto, o patrocinador do usuário pode assinar o seu documento, afirmando assim que o seu escopo reúne as necessidades do negócio.

A modelagem dimensional deveria ser desenvolvida usando-se uma ferramenta que a suportasse (como por exemplo, o *software* Erwin (CA, 2000)). Isso traria uma série de vantagens tais como: integração com outros modelos de dados corporativos, garantia de consistência dos nomes, criação de uma boa documentação, geração automática do projeto físico para os SGBDRs mais populares, etc.

Estudo de caso

Para aplicação do modelo dimensional foi implementado um protótipo de um *data mart* que oferece subsídios e ferramentas gerenciais para a tomada de decisão na gestão de um portal jurídico. Esse portal fornece um completo, eficaz e moderno canal de negócios jurídicos via *internet*, onde profissionais, escritórios e organizações privadas e públicas ligadas ao ramo jurídico têm o seu espaço na economia digital. Configura uma central jurídica na internet.

O portal jurídico está tematicamente organizado e permite o estabelecimento de contatos e prestação de serviços especializados. Tem por objetivo ampliar a visibilidade em âmbito jurídico, redução de custos, maior competitividade, economia de escopo, disponibilidade e agilidade na busca de conhecimento atualizado no local e hora desejados. As consultas de clientes são encaminhadas ao profissional associado mais habilitado a responder sobre o assunto questionado.

O diagrama de entidade-relacionamento, mostrado na Figura 7, se refere aos dados requeridos para a carga do *data mart*. É composto pelas tabelas: Consulta_técnica (onde ficam registrados os dados da consulta feita pelo cliente), Cliente (dados cadastrais de clientes), Área-Técnica (áreas em que está dividida a central jurídica); Unidade_Federação (dados das unidades da federação) e Município (dados dos municípios).

O *data mart* tem por objetivo auxiliar a administração do portal nos seguintes questionamentos:

Quais os Estados ou cidades do Brasil e em que áreas técnicas do direito estão sendo feitas as consultas técnicas em um dado período de tempo? Como direcionar o *marketing* para que ele seja mais eficiente e menos custoso, de modo a expandir o negócio? Como racionalizar os recursos técnicos?



Figura 7. Diagrama de entidade-relacionamento do sistema de origem

A modelagem dimensional do *data mart*, mostrada na Figura 8, para atender aos requisitos citados é composta: pela tabela de fatos Fato_Consulta-Técnica, que contém as consultas por cliente, área técnica e por data. As dimensões são as tabelas: Cliente, Área-Técnica e de tempo *time_by_day*. A tabela Municipio é uma subdimensão da tabela Cliente.

Na etapa do projeto físico foi gerado o Cubo no servidor Olap (On Line Analytical Processing) do gerenciador de banco dados SQL Server 7.0, da Microsoft.

A interface com o usuário (a visualização dos dados a partir do cubo) pode ser feita de duas formas:

1. Por meio da utilização do Olap Browser, do servidor Olap

Nesta ferramenta, pode-se realizar *drill down* e *drill up*, cruzar os dados com as dimensões disponíveis e também filtrar os dados a partir das dimensões, conforme mostrado na Figura 9.

2. Utilização de uma ferramenta *front end* tal como, a planilha Excel da Microsoft

A partir do Excel pode-se conectar uma planilha a um cubo existente. Veja-se o resultado mostrado na Figura 10.

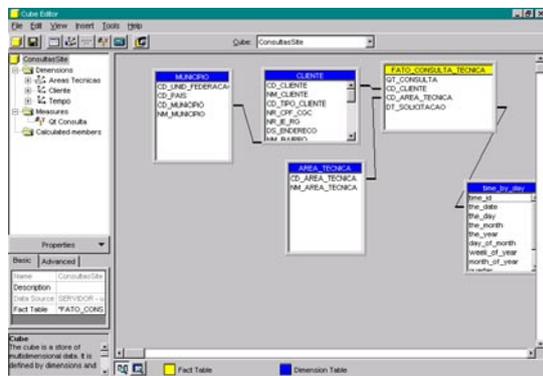


Figura 8. Modelagem dimensional do *data mart*

O gráfico apresentado na Figura 10 é dinâmico, ou seja, sempre que os dados do cubo forem modificados o gráfico muda automaticamente.

Para realização da carga dos dados OLTP (On Line Transaction Processing) para o DW foi utilizada uma ferramenta do SQL Server 7.0 (Data Transformation Services).

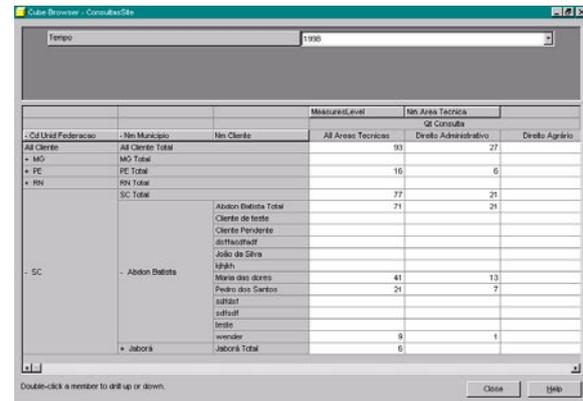


Figura 9. Exemplo de interface com o usuário utilizando o Olap Browser

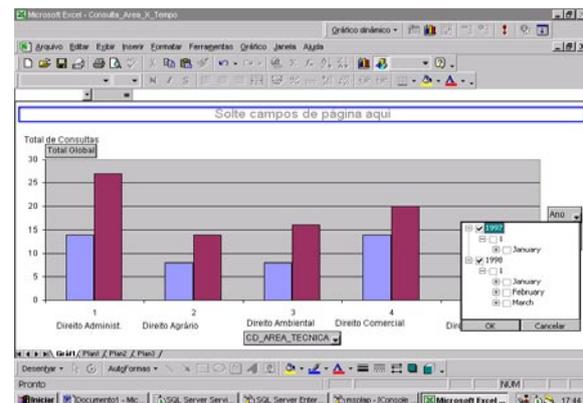


Figura 10. Exemplo de interface com o usuário utilizando a planilha Excel

Considerações finais

O *data warehouse* é o meio e não o fim, pois o fim é o negócio da empresa.

A informação é um produto valioso e confere vantagem competitiva. A modelagem dimensional possui um grande potencial para as aplicações de *data warehouse* e, seus elementos, técnicas e passos que orientam o seu desenvolvimento, aumentam a probabilidade de sucesso do projeto. A criação de um *data warehouse* utilizando-se a modelagem dimensional gera uma interface gráfica simples para a base de dados, permitindo que os usuários entendam e possam desenvolver aplicativos amigáveis para navegar essas bases de modo

eficiente, apoiando significativamente as tomadas de decisões de uma organização.

A existência do modelo E/R dos sistemas fontes do *data warehouse* permite um melhor entendimento dos seus dados, facilitando a modelagem dimensional. A construção do modelo multidimensional deve ser feita sob uma perspectiva do usuário, e não dos dados.

A modelagem dimensional desenvolvida no estudo de caso do portal jurídico, para a construção do DW, revelou-se eficiente para a obtenção de respostas às perguntas inicialmente formuladas. Portanto, o modelo dimensional oferece um potencial significativo e efetivo, para a criação de *data warehouse* com sucesso.

Referências bibliográficas

- CA (Computer Associates International, inc.). Computer Associates Erwin, 2000. URL: <http://ca.com/products/alm/erwin.htm>
- Chen, P. Gerenciando Banco de Dados: A Abordagem Entidade-Relacionamento para Projeto Lógico. McGraw-Hill, 1990.
- Daphne Corp. Building Corporate Business Intelligence: From Data marts to the Enterprise Data Warehouse, 1999. URL: <http://www.daphnesoft.com/whitepaper.doc>
- Freitas, A, A “Data Mining”- Mineração de Dados. Cefet Pr - Nucleo, 1998.
- Harrison, T.H. Intranet data warehouse, Berkeley, 1998.
- Inmon, W.H. What is a data warehouse?, 1995 URL: http://www.cait.wustl.edu/papers/prism/vol1_no1/
- Kern, M.V. Modelagem da Informação com IDEF1X: Linguagem, Método, Princípio do Consenso. *Rev. Alance*, 3:99-107, 1999.
- Kimbal, R. *et al.* The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses. John Wiley & Sons, New York, 1998.
- Meyer, D. Building a better data warehouse, Prentice Hall, 1998.
- Nist (National Institute of Standards and Technology). Federal Information Processing Standards Publication 184. Integration Definition for Information Modeling (IDEF1X). Gaithersburg, MD (USA), December 1993.
- Teorey, J.T. Database Modeling & Design. Morgan Kaufmann Publishers, San Francisco, 1999.

Received on October 07, 2000.

Accepted on November 22, 2000.