



[Tecnologias & Produtos](#) > [Arquitetura](#)

Fundamentos e Modelagem de Bancos de Dados Multidimensionais

Publicado em: 4 de outubro de 2007
Por Alexandre Ricardo Nardi

-Arquiteto de Soluções

Microsoft Brasil

Conteúdo

Nesta página

- ↓ [Introdução](#)
- ↓ [Fundamentos de Sistemas Analíticos](#)
- ↓ [Sistemas Transacionais X Sistemas Analíticos](#)
- ↓ [Data Warehouses e Data Marts](#)
- ↓ [Bancos de Dados Multidimensionais](#)
- ↓ [Modelos de Dados Multidimensionais](#)
- ↓ [Alguns Conceitos](#)
- ↓ [Estrela e suas Variações](#)
- ↓ [Snowflake e suas Variações](#)
- ↓ [Conclusão](#)
- ↓ [Referências](#)

Resumo: Este artigo apresenta estruturas de indexação para bancos de dados multidimensionais, as Árvore Bitmap, comparando com as estruturas usadas para bancos de dados relacionais. Então percorremos diversas opções para modelagem de dados multidimensionais, incluindo os Modelos Estrela e Snowflake e suas variações.



Introdução

A utilização de Sistemas Gerenciadores de Bancos de Dados Relacionais (SGBDRs) é prática consolidada mundialmente. Os dados precisam poder ser armazenados e recuperados geralmente em intervalos curtos de tempo, em situações não triviais tais como:

- Situações de alta concorrência, por vezes de milhares de acessos simultâneos, que precisam ser gerenciados em ambientes transacionais;
- Grandes web sites que, além dos requisitos de desempenho, necessitam cuidados especificamente com relação à segurança dos dados;
- Aplicações analíticas, baseadas em históricos de anos, para apoio a decisões gerenciais e estratégicas.

As estruturas de dados e mecanismos de indexação utilizados por esses SGBDRs atendem bem às duas primeiras situações. Todavia, as aplicações analíticas possuem peculiaridades tais como manipulação de grandes volumes de dados e baixa taxa de atualização. Essas características favorecem outro modelo estrutural, mais eficiente e por vezes mais econômico, no tocante a espaço de armazenamento.

Conhecendo as estruturas de dados, a compreensão das alternativas de modelagem de dados multidimensional, base dos sistemas analíticos, fica facilitada.

Neste texto abordaremos as principais diferenças entre sistemas transacionais e analíticos, bem como algumas estruturas de indexação comumente utilizadas para cada tipo. Apresentaremos, ainda, considerações importantes quanto à modelagem de dados multidimensional, incluindo os modelos estrela e snowflake e suas variações.

↶ [Início da página](#)

Fundamentos de Sistemas Analíticos

Nos últimos anos o termo Business Intelligence (BI) tem sido largamente utilizado no mercado como sinônimo de sistemas analíticos, OLAP, cubos, entre outros. Embora essas denominações possam estar associadas entre si, são conceitualmente

distintas.

A rigor, Business Intelligence pode ser obtido por qualquer artefato, seja tecnológico ou não, que permita a extração de conhecimento a partir de análises do negócio. Por razões óbvias, a efetividade destas análises será maior se os dados estiverem disponíveis de modo consistente e, preferencialmente, consolidado. Este é um dos objetivos dos Data Warehouses.

Soluções informatizadas de BI geralmente contém sistemas analíticos, que podem ser de diversos tipos, dependendo do objetivo das análises e do perfil do usuário, conforme ilustrado na Figura 1:

- Decision Support Systems (DSS), ou Sistemas de Apoio a Decisão: são baseados em relatórios analíticos, normalmente utilizados por usuários de nível operacional;
- Management Information Systems (MIS), ou Sistemas de Informações Gerenciais: permitem análises mais profundas, com a realização de simulações de cenários. Por vezes, utilizam-se de ferramentas de Data Mining para identificação de cruzamentos não triviais. São utilizados por analistas de negócio no nível tático;
- Executive Information Systems (EIS), ou Sistemas de Informações Executivas: são voltados para profissionais que atuam no nível estratégico das empresas, como diretores e presidência. Oferecem, para tanto, um conjunto de indicadores chave de desempenho (KPI, ou Key Performance Indicators).

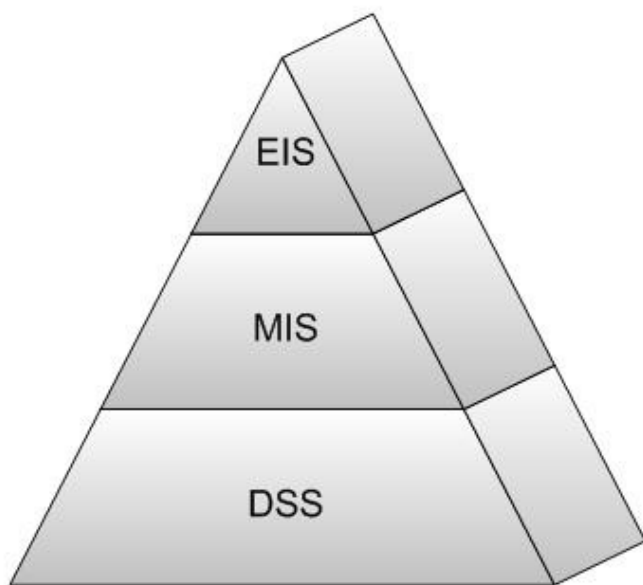


Figura 1: alguns tipos de sistemas analíticos.

O afinamento da pirâmide indica quantidades menores e mais específicas de usuários para cada sistema.

Independentemente do tipo de sistema analítico, este difere substancialmente dos sistemas transacionais de produção. A seguir, apresentaremos tais diferenças, bem como os conceitos envolvidos.

[↗ Início da página](#)

Sistemas Transacionais X Sistemas Analíticos

Sistemas transacionais, também conhecidos como sintéticos ou ainda OLTP – Online Transactional Processing - são aqueles que, como o nome sugere, baseiam-se em transações. Alguns exemplos deste tipo de sistemas são:

- Sistemas Contábeis;
- Aplicações de Cadastro;
- Sistemas de Compra, Estoque, Inventário;
- ERPs, CRMs.

Os sistemas transacionais se caracterizam pela alta taxa de atualização, grande volumes de dados e acessos pontuais, ou seja, pesquisas cujo resultado seja de pequeno volume (até milhares de linhas, mas preferencialmente menos).

Já os sistemas analíticos, ou OLAP – Online Analytical Processing – se caracterizam por fornecer subsídio para tomadas de decisão, a partir de análises realizadas sobre bases de dados históricas, por vezes com milhões de registros a serem totalizados. Alguns exemplos de sistemas analíticos são os ilustrados na Figura 1.

A Tabela 1 sintetiza as principais diferenças entre sistemas transacionais e analíticos:

Tabela 1: Comparação entre sistemas transacionais e analíticos.

<u>Característica</u> <input type="checkbox"/>	<u>Sistemas Transacionais(OLTP)</u> <input type="checkbox"/>	<u>Sistemas Analíticos(OLAP)</u> <input type="checkbox"/>
Atualizações	Mais freqüentes	Menos freqüentes
Tipo de Informação	Detalhes	Agrupamento
Quantidade de Dados	Poucos	Muitos
Precisão	Dados atuais	Dados históricos
Complexidade	Baixa	Alta
Consistência	Microscópica	Global
Exemplos	CRM, ERP, Supply Chain	MIS, DSS, EIS
Terminologia	Linhas e Colunas	Dimensões, Medidas e Fatos

Conforme ilustrado na tabela acima, o fato dos sistemas transacionais refletirem a situação atual de um determinado tipo de dado conduz todas as demais características, como:

- A realização de atualizações freqüentemente, mantendo os dados atuais;
- Informação detalhada com a maior granularidade possível (consistência microscópica);
- Pesquisas pontuais, portanto de baixa complexidade, no tocante ao negócio (do ponto de vista técnico, a pesquisa pode ser bem elaborada).

Do mesmo modo, o fato das análises serem realizadas sobre dados históricos leva às seguintes características:

- Uma vez que os dados são históricos, as atualizações não precisam ser tão freqüentes. Por exemplo, numa comparação entre a produtividade de três filiais de uma empresa para um determinado produto nos últimos quatro meses, por mês, o dia de hoje ou mesmo ontem não é, em geral, de grande representatividade;
- As análises geralmente agrupam informações, sendo tais agrupamentos mais importantes neste contexto do que os dados detalhados. No exemplo do item anterior, o importante é a produção conjunta mensal, e não a produção de uma unidade particular do produto analisado.

Os diferentes tipos de sistemas também sugerem diferentes abordagens técnicas, seja na forma de armazenamento ou de busca. No caso dos sistemas transacionais, que exigem acesso rápido aos dados, principalmente no tocante a modificações, a utilização de índices estruturados como árvores balanceadas, ou B-Trees, é adequada. No entanto, essa estrutura não é a mais recomendada para sistemas analíticos, onde as atualizações são esporádicas, mas as consultas envolvem grandes conjuntos de dados e devem ser muito rápidas. Outra estrutura de dados, denominada árvore PATRICIA, é mais adequada neste contexto. As próximas duas seções descrevem o funcionamento dessas duas estruturas.

Árvores Balanceadas (B-Trees)

Este tipo de árvore, empregada por vários SGBDRs comerciais ou não, possui a denominação balanceada pelo fato das folhas estarem praticamente à mesma distância da raiz, podendo diferir em apenas um nível. A Figura 2 ilustra uma árvore balanceada:

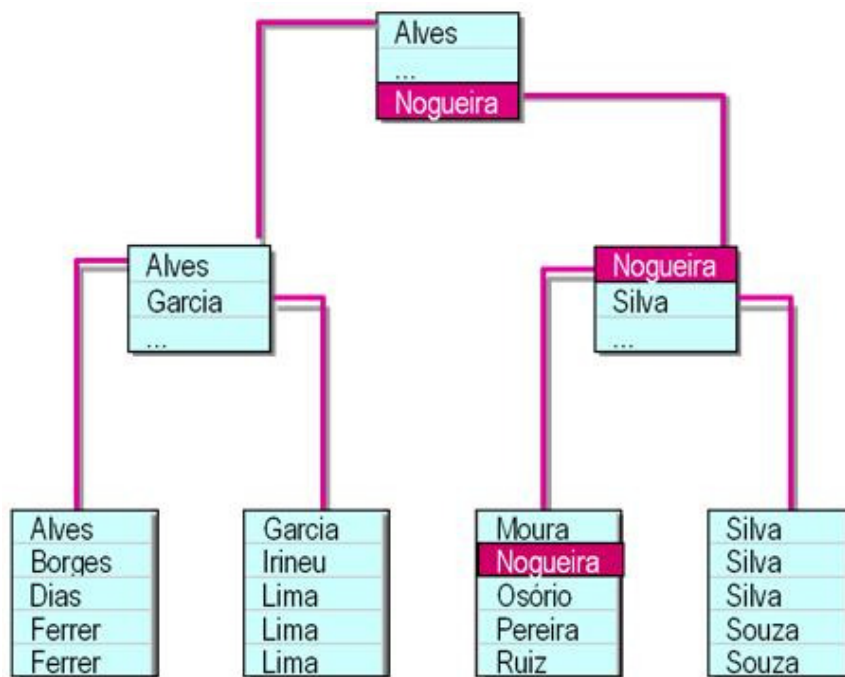


Figura 2: exemplo de árvore balanceada.

A figura acima mostra uma busca na árvore, que pode ser realizada de modo eficiente. O fato da árvore acima ser binária é mera coincidência. Vale notar que uma inserção neste tipo de árvore pode ser realizada com pequena quantidade de operações, conforme ilustrado na Figura 3. Essas operações denominam-se rotações, e visam a manutenção da árvore como balanceada. Nos dois casos, a inserção foi realizada em g, de modo a desbalancear a árvore. As rotações sugeridas reparam a situação.

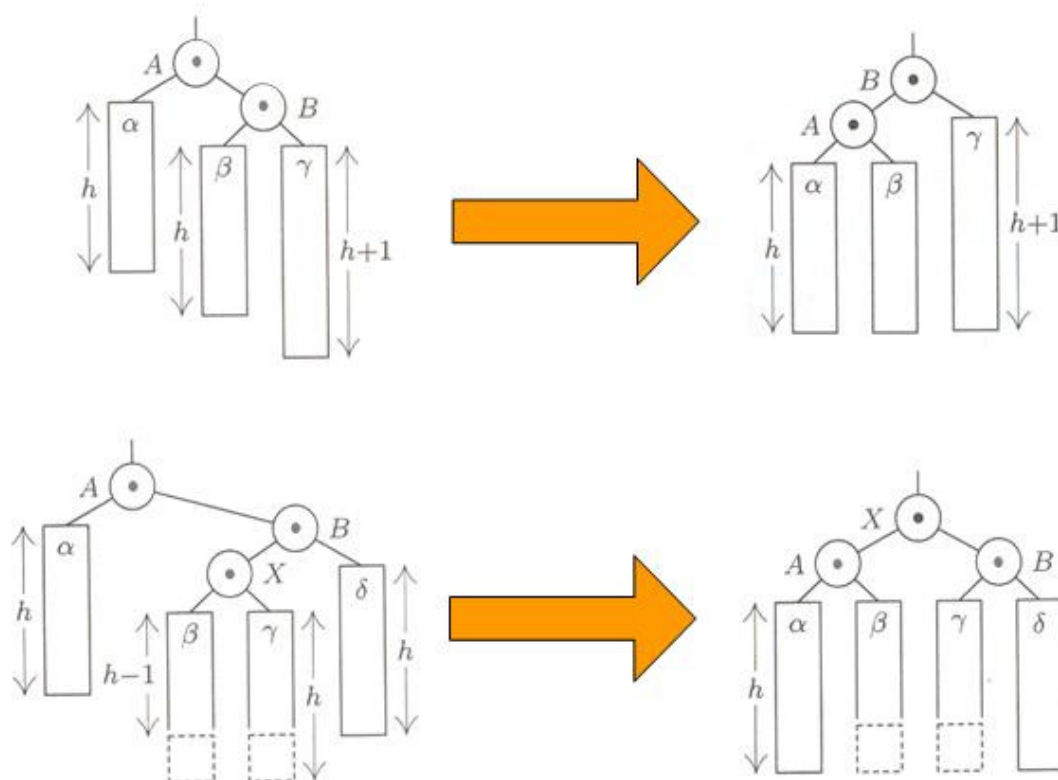


Figura 3: inserções em árvores balanceadas.

Árvores PATRICIA

Concebido por Donald. R. Morrison (1) e descrito em (2), PATRICIA é um algoritmo para realização de buscas em árvores com as chaves dos nós representadas em binário, sem armazenar as chaves nos nós. O nome é um acrônimo de "Practical Algorithm To Retrieve Information Coded In Alphanumeric", e o método é particularmente útil para tratamento de chaves de tamanho variável extremamente longas, tais como títulos e frases. No caso de pesquisas analíticas, os dados podem tirar proveito deste método desde que as informações sejam armazenadas como cadeias de texto.

Uma restrição dessas árvores é a necessidade de não haver um elemento que seja prefixo de outro, o que pode facilmente ser obtido se necessário.

Em nosso exemplo, utilizaremos a codificação apresentada na Tabela 2 para as letras de "A" a "Z":

Tabela 2: representação binária das letras do alfabeto.

<u>Caractere</u> <input type="checkbox"/>	<u>Representação Decimal</u> <input type="checkbox"/>	<u>Representação Binária</u> <input type="checkbox"/>
<espaço>	0	00000
A	1	00001
B	2	00010
C	3	00011
D	4	00100
E	5	00101
F	6	00110
G	7	00111
H	8	01000
I	9	01001
J	10	01010
K	11	01011
L	12	01100
M	13	01101
N	14	01110
O	15	01111
P	16	10000
Q	17	10001
R	18	10010
S	19	10011
T	20	10100
U	21	10101
V	22	10110
W	23	10111
X	24	11000
Y	25	11001
Z	26	11010

O caractere <espaço> é utilizado como separador entre as palavras.

A Tabela 3 ilustra a representação binária da frase "ESTE FOI UM ESTUDO DE ARVORE PATRICIA":

Tabela 3: representação em binário das palavras do exemplo.

Palavra	Representação							
ESTE	E 00101	S 10011	T 10100	E 00101				
FOI	F 00110	O 01111	I 01001					
UM	U 10101	M 01101						
ESTUDO	E 00101	S 10011	T 10100	U 10101	D 00100	O 01111		
DE	D 00100	E 00101						
ARVORE	A 00001	R 10010	V 10110	O 01111	R 10010	E 00101		
PATRICIA	P 10000	A 00001	T 10100	R 10010	I 01001	C 00011	I 01001	A 00001

A partir da frase acima é possível construir a árvore PATRICIA ilustrada na Figura 4:

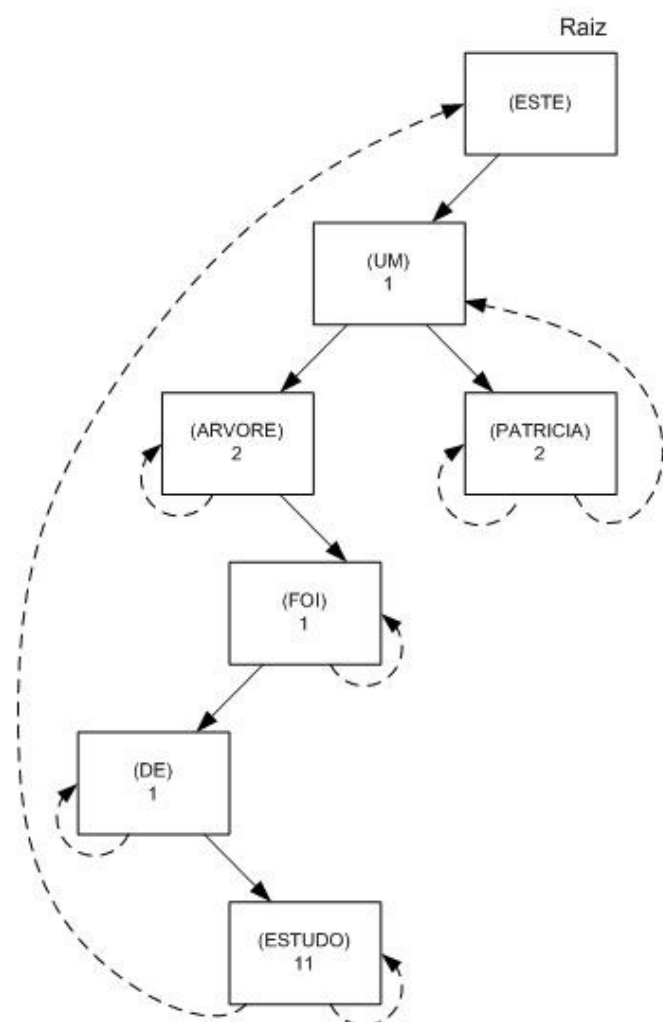


Figura 4: árvore PATRICIA.

As setas tracejadas indicam um ancestral do nó corrente, e as setas Cheias indicam seus descendentes.

Cada nó da árvore possui as seguintes informações:

- KEY: um ponteiro para o início da palavra no texto original. Por exemplo, ao invés do texto "(ESTUDO)", o nó deveria conter 13, que é a posição na frase do início da palavra;

- LLINK e RLINK: ponteiros representando zero e um, respectivamente. Mais detalhes no exemplo de busca na árvore a seguir;
- LTAG e RTAG: campos binários indicando se LLINK e RLINK, nesta ordem, são ponteiros para nós ancestrais (valor um, representado pelas setas tracejadas na Figura 4) ou descendentes (valor zero, correspondente às setas cheias na Figura 4). Por convenção, um nó pode ser ancestral de si mesmo;
- SKIP: o número de bits que podem ser pulados durante uma busca. O exemplo de busca a seguir ilustra esse conceito.

A raiz da árvore contém apenas KEY, LLINK e LTAG.

Busca em Árvores PATRICIA

O algoritmo de busca é relativamente simples. Ilustraremos aqui dois casos:

1. Busca por um elemento presente na árvore: procura pela palavra "FOI" (em binário, 00110 01111 01001)
 - A busca inicia no nó à esquerda da raiz, "(UM)". O campo SKIP pede para analisarmos o bit 1. Se valer 0, procura à esquerda. Se valer 1, procura à direita. No exemplo, seguimos para a esquerda;
 - Estando no nó "(ARVORE)", o campo SKIP = 2 somado aos SKIPS anteriores, ou 1 + 2, pede para analisar o bit 3 do texto sendo procurado. Como vale 1, procura à direita;
 - Estando no nó "(FOI)", o campo SKIP pede para analisar o bit 4. Como vale 1, segue à direita;
 - Como o nó corrente não mudou, testa o valor da chave para saber se encontrou. Esse teste é importante, uma vez que o padrão "0x11x xxxxx xxxxx" foi encontrado. Uma busca pelo texto "FUI" também terminaria no mesmo padrão.
2. Busca por todos os elementos que começam com uma cadeia. Por exemplo, busca por todos os elementos que comecem por "EST" (em binário, 00101 10011 10100):
 - A busca procede da mesma forma que a anterior, até o momento em que tenta-se comparar bit inexistente (16º bit) no texto procurado (15 bits);
 - Nesse ponto, compara-se o texto procurado com o elemento corrente - "(ESTUDO)". Se encontrou, então o mesmo padrão também está presente nos descendentes e ancestrais deste nó, devendo ser então comparados.

Construção de Árvores PATRICIA

Este algoritmo funciona da seguinte forma:

1. A raiz da árvore contém a primeira palavra lida, conforme a Figura 5;
2. A partir daí, para cada novo elemento, fazemos duas buscas:
 - A primeira para procurar a provável localização do elemento. Considerando todos os elementos distintos, essa busca sempre será mal-sucedida;
 - Na posição encontrada na primeira busca, determina-se quantos bits coincidem com o elemento novo;
 - Daí, faz-se nova busca para encontrar precisamente a posição a inserir o novo elemento.

A sequência de figuras a seguir ilustra a construção da árvore da Figura 4:



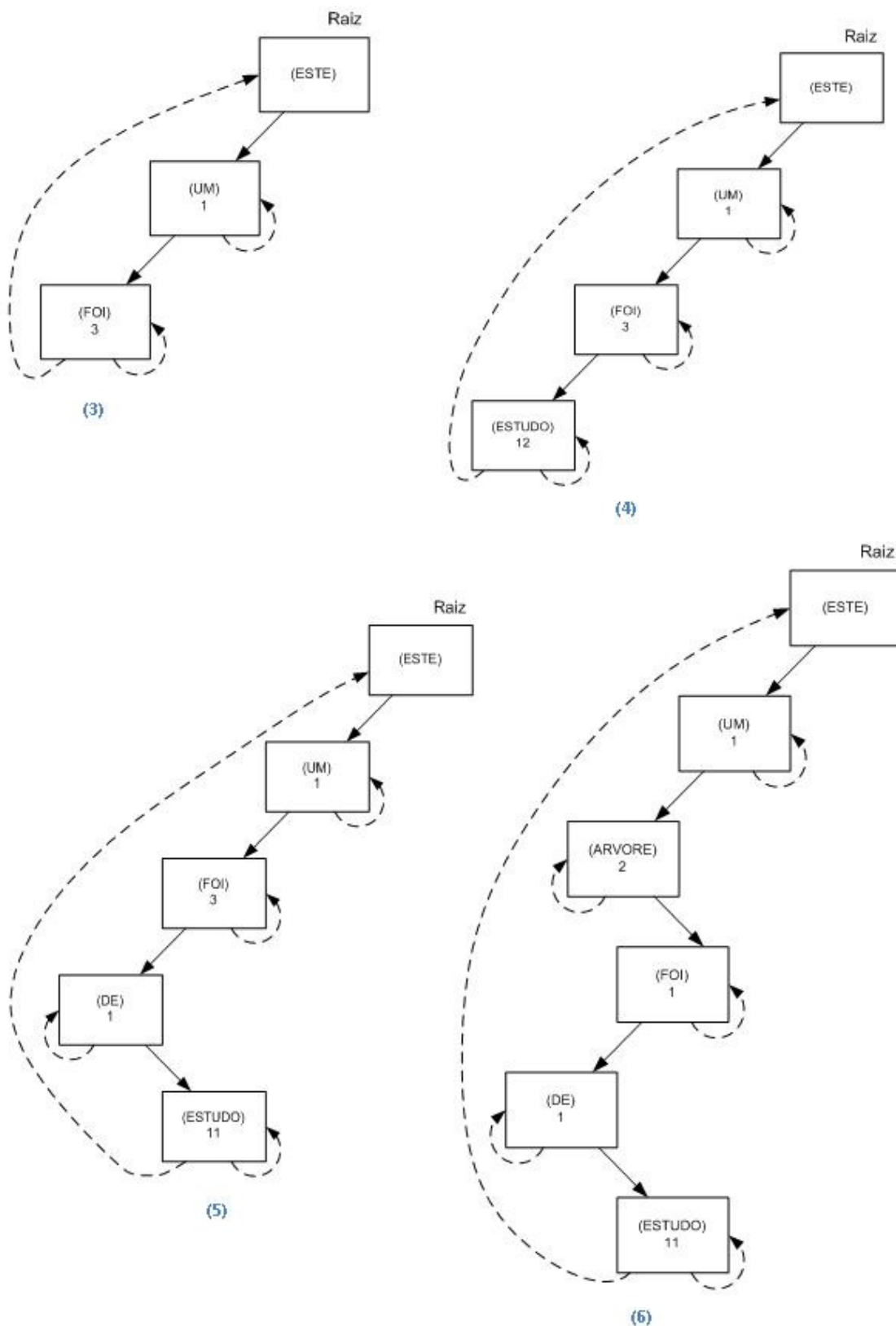


Figura 5: sequência de figuras ilustrando a construção de uma árvore PATRICIA.

A árvore completa está na Figura 4.

Embora seja uma estrutura de dados muito eficiente no que diz respeito a buscas, executando em tempos $O(\lg N)$, onde N é o número de elementos da árvore, o custo de atualizações é grande, inviabilizando seu uso para aplicações transacionais.

[↗ Início da página](#)

Data Warehouses e Data Marts

Esses dois conceitos comumente se associam ao uso de aplicações analíticas.

Um Data Warehouse é uma base de dados, geralmente relacional, que consolida as informações empresariais. Sua

construção (3) é um processo normalmente moroso e complexo, por diversos fatores, dentre os quais a grande quantidade de dados, diversas fontes de informações com bases heterogêneas e muitas vezes inconsistentes, envolvimento de várias áreas ou departamentos da empresa. Um dos maiores desafios na construção do DW é a extração e consolidação dos dados operacionais, pois:

- Pode haver várias fontes;
- Os dados precisam ser limpos;
- A granularidade precisa ser ajustada;
- Pode ser necessário resumir dados;
- Deve haver valores default e tratamento de NULL;
- É necessário componente temporal;
- Os relacionamentos nos dados de entrada precisam ser claros.

Algumas situações comuns entre as fontes de dados:

- Mesmos dados, nomes diferentes;
- Dados diferentes, mesmo nome;
- Dados exclusivos de uma aplicação;
- Chaves diferentes, mesmos dados.

Como métodos de construção, existem formalmente dois:

- Top-down, no qual é realizada a modelagem integral do DW, seguida pelas extrações de dados. A principal vantagem é a criação de um modelo único. O revés fica por conta do maior tempo de projeto;
- Bottom-up, onde o foco é em uma área por vez, com o crescimento gradual do DW. A vantagem é a obtenção de resultados a intervalos mais curtos, garantindo muitas vezes sustentação ao projeto. A desvantagem é a maior dificuldade de se consolidar informações entre as diversas áreas.

Uma alternativa às estratégias acima, denominada Middle-out, é aproveitar as vantagens de cada uma por meio do desenvolvimento iterativo do DW:

1. O modelo de dados corporativo é o primeiro a ser desenvolvido e é o responsável pela integração dos demais;
2. As primeiras tabelas da área de interesse escolhida são povoadas: primeiras análises;
3. Povoamento de mais tabelas com dados históricos;
4. Alguns dados passam a compor o DW, saindo da base operacional;
5. Surgimento dos data marts (a seguir nesta seção);
6. O ciclo se repete até que o DW esteja completo. Bases de produção contêm apenas dados operacionais.

Outro fator crítico para o sucesso de um DW é o gerenciamento do volume. Embora o conceito de DW se aplique a grandes quantidades de dados, chegando atualmente a ordem de TB, sua capacidade não é infinita, devendo ser utilizada sabiamente. Apenas dados relevantes deveriam constar do DW. Pode ser que o horário de uma determinada transação seja importante quando o foco for o curto prazo, mas que apenas um contexto de agrupamento seja suficiente para dados de cinco anos atrás. Questões como essa devem ser consideradas durante o planejamento do DW, pois ajudam a dimensioná-lo.

A remoção de dados do DW é um assunto tratado com receio pelos DBAs e pelos analistas de negócio. A rigor, tão importante quanto saber que dados armazenar, é saber quando e quais dados remover do DW. Algumas estratégias são:

- resumir dados mais antigos;
- Armazenar os dados antigos em meio mais barato (fita);
- Remover os dados do DW.

Tais estratégias não são excludentes, podendo ser utilizadas em conjunto.

A importância da remoção de dados está em manter o DW o mais enxuto possível, embora isso possa parecer contraditório ao conceito de DW.

Com relação à granularidade, as bases de dados operacionais trabalham com o maior nível de detalhe possível, ou seja, a

maior granularidade. Já no DW pode haver diversos graus de agregação e resumo dos dados. Por exemplo, os dados do ano corrente podem ser detalhados por item de pedidos, de um a cinco anos, por total de cada pedido e, após isso, por total de pedidos por dia. A correta determinação da granularidade exerce papel fundamental no planejamento de capacidade e desempenho do DW.

Ao contrário do que ocorre com as bases operacionais, o DW, por conter dados históricos, não demanda alta taxa de atualização. Desse modo, pode ser atualizado a cada 24 horas ou até mesmo uma vez por semana. Além disso, por sofrer poucas modificações, e de forma controlada (por aplicações específicas para esse fim), seus relacionamentos podem ser implementados através de entidades, embora isso não seja frequente.

Embora consolide as informações da empresa, mesmo que de modo iterativo, os DWs são geralmente armazenados em bancos de dados relacionais, e não se utilizam de estruturas tais como as árvores PATRICIA. Neste ponto surgem os Data Marts (DM), bancos de dados multidimensionais específicos por área de negócio para realização de análises. Alguns conceitos sobre Data Marts não estão muito bem claros para o mercado. O processo de construção de um DW e de DMs, ilustrado na Figura 6, ajuda a esclarecer alguns deles:

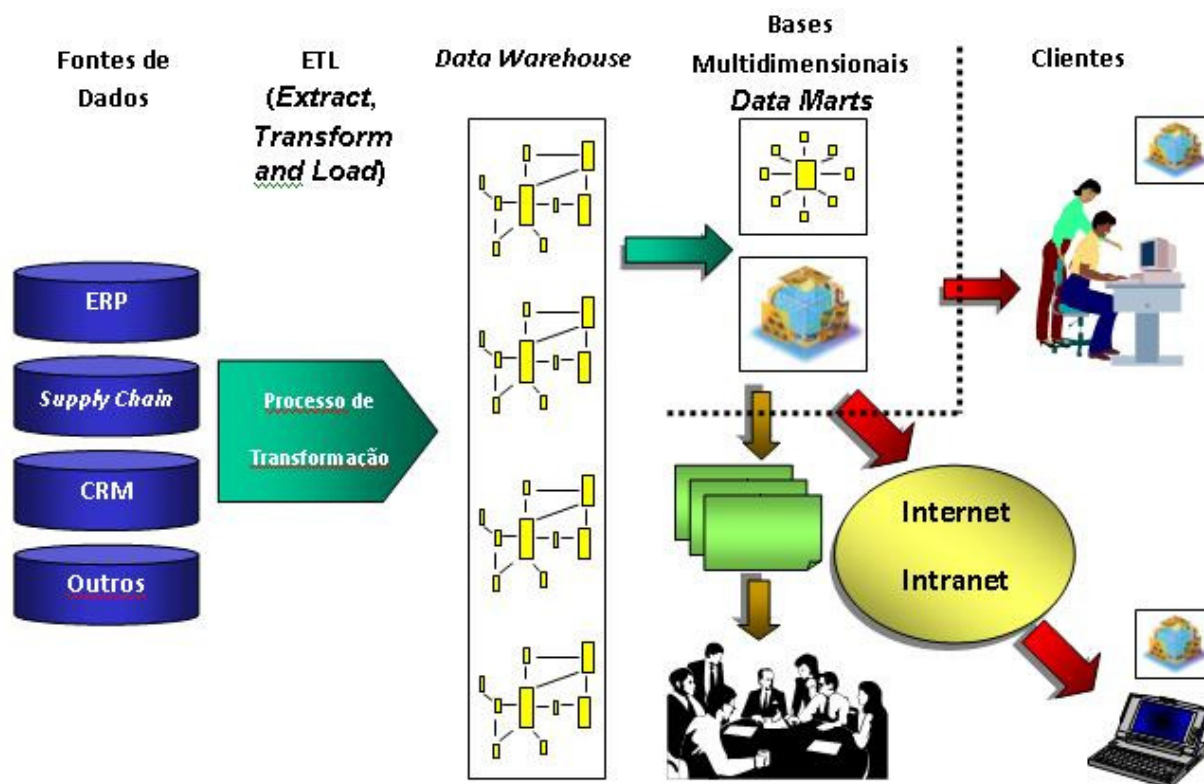


Figura 6: etapas da construção de um DW e de DMs.

Como pode ser notado na figura acima, um DW construído iterativamente possui porções agrupadas por segmento de negócio, região ou qualquer outra forma que seja adequada à empresa. Essas porções alimentam os Data Marts, que podem ser então consultados por ferramentas de análise.

[↗ Início da página](#)

Bancos de Dados Multidimensionais

A finalidade de bases de dados multidimensionais (alguns autores chamam de dimensionais) é fornecer subsídio para realização de análises. Para tanto, sua arquitetura e até mesmo a terminologia empregada são distintas das utilizadas para bancos de dados transacionais.

O fato de existirem diversas informações a serem cruzadas (dimensões) permite a realização de pesquisas tais como a ilustrada na Figura 7:



Figura 7: exemplo de pesquisa multidimensional.

Terminologia

As análises sobre dados históricos envolvem uma série de possibilidades de cruzamentos e agrupamentos de informações, com o uso dos seguintes termos:

- **Dimensões:** estabelecem a organização dos dados, determinando possíveis consultas/cruzamentos. Por exemplo: região, tempo, canal de venda,... Cada dimensão pode ainda ter seus elementos, chamados membros, organizados em diferentes níveis hierárquicos. A dimensão tempo, por exemplo, pode possuir duas hierarquias: calendário gregoriano (com os níveis ano, mês e dia) e calendário fiscal (com os níveis ano, semana e dia);
- **Medidas:** são os valores a serem analisados, como médias, totais e quantidades;
- **Fatos:** são os dados a serem agrupados, contendo os valores de cada medida para cada combinação das dimensões existentes. O tamanho da tabela que contém os fatos merece atenção especial do analista;
- **Agregações:** totalizações calculadas nos diversos níveis hierárquicos.

A criação de DMs implica na geração de agregações. Este processamento se reflete em ganho de desempenho quando da realização de consultas.

Alicerce Relacional

Diversas ferramentas analíticas, também chamadas ferramentas de OLAP, operam sobre bases de dados multidimensionais armazenadas em SGBDRs. Além disso, as agregações são também mantidas em banco de dados relacional.

Esta forma de armazenamento é conhecida como ROLAP, ou Relational OLAP. Uma vez que os dados já se encontram em um modelo apropriado, chamado multidimensional (veja opções de modelagem nas próximas seções), basta processar as agregações. Com isso obtém-se ganho de espaço de armazenamento, uma vez que os dados permanecem apenas na base de origem (multidimensional), embora a criação de grandes quantidades de agregações possa incorrer em explosão de dados.

Alicerce em Cubos

Outra forma de armazenamento, cujo modelo matemático denomina-se hipercubos, apresenta a característica de possuir armazenamento e indexação em estruturas de dados que otimizam consultas ao invés de atualizações, como é o caso das árvores PATRICIA.

Esta forma é erroneamente chamada MOLAP, ou Multidimensional OLAP. O erro está no fato de que bases ROLAP também são multidimensionais.

Quando o modelo multidimensional é processado, nova base é gerada, desta vez contendo tanto os dados quanto as agregações em formato próprio, utilizando-se de estruturas apropriadas para pesquisas.

A Figura 8 ilustra uma representação de um cubo com três dimensões:

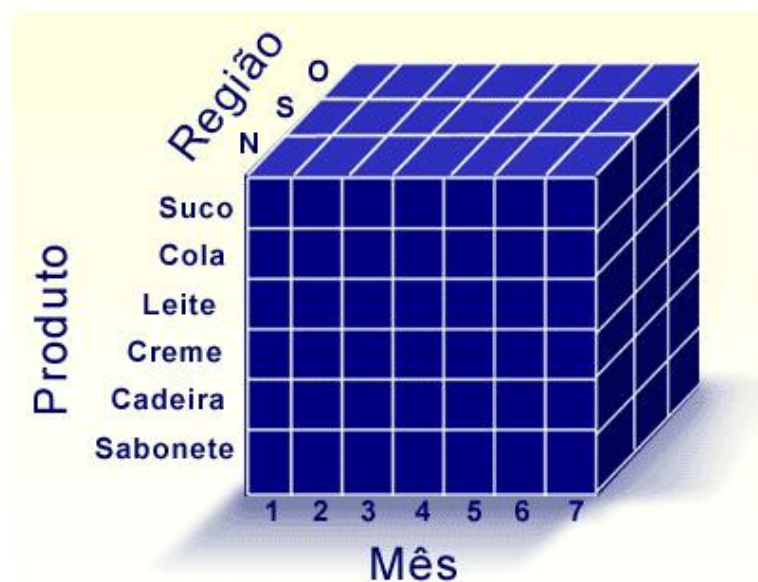


Figura 8: representação de um cubo com as dimensões Produto, Região e Tempo.

Embora o risco de explosão de dados seja comum em estruturas relacionais por conta de cruzamentos sem dados, mas que ocupam algum espaço, as estruturas utilizadas pelos cubos são esparsas, e se aplicam a dados e agregações, de modo que os cubos são substancialmente menores do que a base multidimensional que o originou.

[↗ Início da página](#)

Modelos de Dados Multidimensionais

A natureza do uso de bancos de dados multidimensionais torna sua modelagem distinta daquela utilizada para sistemas transacionais. Neste último aplicamos técnicas de normalização seguidas por graus de desnormalização a fim de obter o desempenho desejado ao reduzir o número de tabelas em junções (joins).

Vale lembrar que o número de planos de execução para uma junção de n tabelas é $n!$, isto é, para uma junção de 10 tabelas há 3.628.800 possibilidades. Embora o escalonador do SGBD possua estratégias para reduzir este número, é um ponto de atenção a considerar. Já para o caso dos MDDBs, o grau de desnormalização é bem maior, dado o volume de dados e a agilidade na consolidação de valores quando calculando as agregações.

Nesta seção, com trechos extraídos de (4) e (5), percorremos alguns conceitos importantes para a modelagem quanto à representação de fatos, dimensões e quanto a chaves. Então descrevemos vários modelos de dados, sempre do ponto de vista lógico. Portanto, os modelos que veremos serão sempre relacionais, independentemente do alicerce, relacional ou em cubos, que pode ser utilizado para o modelo físico.

[↗ Início da página](#)

Alguns Conceitos

Quando o modelo de dados começa a ser definido, elementos básicos de representação precisam ter sido estabelecidos, de modo a criar-se um padrão de modelagem. Em nosso modelo teremos as dimensões e fatos representados em tabelas, podendo haver múltiplas dimensões e múltiplas tabelas de fatos.

Fatos

Ao modelar a(s) tabela(s) de fatos (ou apenas tabela fato), deve-se ter em mente os seguintes pontos:

- A chave primária é composta, sendo um elemento da chave para cada dimensão;
- Cada elemento chave para a dimensão deve ser representado e descrito na "tabela dimensão" correspondente (para efetuar a junção);
- A dimensão tempo é sempre representada como parte da chave primária.


Dimensões

Deve haver uma "tabela dimensão" para cada dimensão do modelo, contendo:

- Uma chave artificial (ou gerada) genérica;
- Uma coluna de descrição genérica para a dimensão;
- Colunas que permitam efetuar os filtros;

- Um indicador NÍVEL que indica o nível da hierarquia a que se refere a linha da tabela.

A Figura 9 ilustra uma tabela para a dimensão "Geografia", com os pontos acima representados. Note que a coluna "nível" determina a hierarquia (Região/Estado/Cidade)



		loja	mercado	região	nível
1001	Vitória	101	10	1	1
1002	Vila Velha	102	10	1	1
1003	Guarapari	103	10	1	1
1004	São Paulo	104	20	1	1
1005	Santos	105	20	1	1
1006	Rio de Janeiro	106	30	1	1
1007	Angra dos Reis	107	30	1	1
1008	Salvador	108	40	2	1
1009	Porto Seguro	109	40	2	1
1010	Espírito Santo		10	1	2
1011	São Paulo		20	1	2
1012	Rio de Janeiro		30	1	2
1013	Bahia		40	2	2
1014	Sudeste			1	3
1015	Nordeste			2	3

Chave da Dimensão

Figura 9: exemplo de uma tabela de dimensão

Valores nulos irão existir em algumas colunas, dependendo do nível hierárquico para o qual a linha contenha valores. Esse é o caso da coluna "loja": como somente existem lojas nas cidades, e não nos estados ou regiões, a tabela fica com nulos, conforme identificados pela região circundada na figura. Todavia, é tarefa do modelo físico reduzir o espaço ocupado pelos nulos

A Dimensão Tempo

Esta é uma dimensão que praticamente todos os sistemas analíticos possuem, dada a característica de realização de análises em dados históricos. Deveria conter:

- Uma coluna chave para a junção com a(s) tabela(s) de fato(s);
- Uma descrição genérica para cada período;
- Colunas que permitam efetuar os filtros;
- Coluna sinalizadora da presença de fatos para o período de tempo indicado na linha;
- Coluna RESOLUÇÃO usada para restringir o período ao nível apropriado - opera de forma idêntica à coluna NÍVEL das outras dimensões
- Coluna SEQÜÊNCIA que contém um número seqüencial de 1 a n em cada nível do período de tempo e identifica a ordem relativa de cada data. Permite:
 - Coluna SEQÜÊNCIA que contém um número seqüencial de 1 a n em cada nível do período de tempo e identifica a ordem relativa de cada data. Permite:
 - Construções com cálculos de tempo, como "últimos quatro dias", por exemplo.

A Figura 10 mostra um exemplo de tabela de dimensão tempo. Note que a descrição é o que aparecerá para os valores de uma determinada data ou período.

TempoID	Descricao	Ano	Trimestre	Mes	HaFatos	Resolucao	Sequencia
950715	Julho 15, 1995	1995	terceiro	julho	S	diaria	1
950716	Julho 16, 1995	1995	terceiro	julho	S	diaria	2
950717	Julho 17, 1995	1995	terceiro	julho	N	diaria	3
950718	Julho 18, 1995	1995	terceiro	julho	S	diaria	4
950719	Julho 19, 1995	1995	terceiro	julho	S	diaria	5
950720	Julho 20, 1995	1995	terceiro	julho	S	diaria	6
9506	Junho, 1995	1995	segundo	junho	S	mensal	1
9507	Julho, 1995	1995	terceiro	julho	S	mensal	2
9508	Agosto, 1995	1995	terceiro	agosto	N	mensal	3

Figura 10: exemplo de uma tabela para a dimensão tempo**Considerações sobre Chaves**

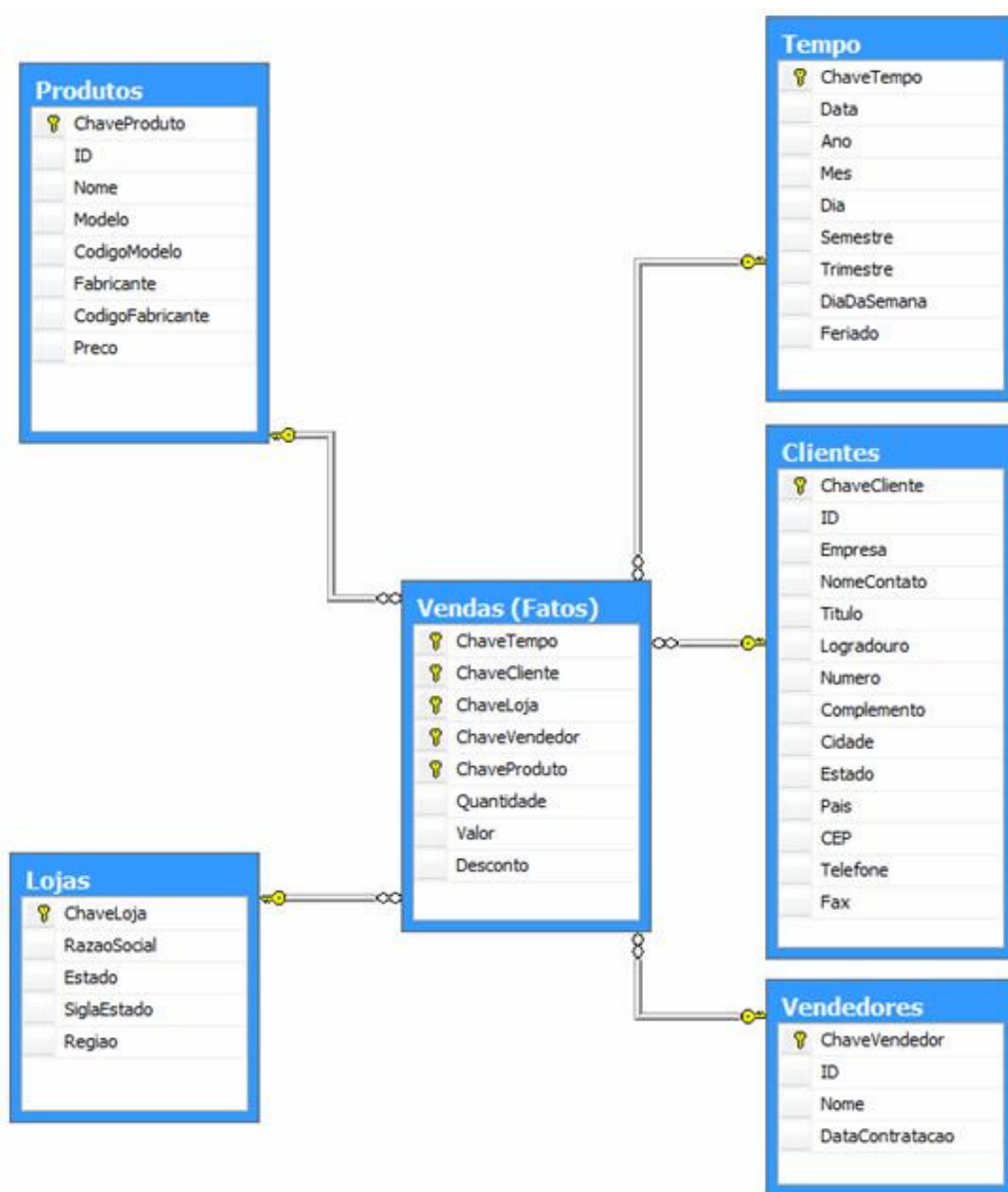
No tocante às chaves, sistemas analíticos devem contar com chaves artificiais, por uma série de motivos:

- Qualquer atualização de dados fica simplificada. Por exemplo, um cadastramento de CPFs, embora improvável, poderia resultar em atualização de grande volume para uma tabela de fatos de transações bancárias, caso o cliente fosse identificado com CPF sendo chave;
- Com uma única coluna para a chave, geralmente de tipo inteiro, o desempenho de pesquisas tende a ser melhor – quanto menor o tamanho da chave, melhor o desempenho;
- O fato de ser chave simples facilita a execução de junções.

[↑ Início da página](#)

Estrela e suas Variações

Uma das formas de apresentação de um banco de dados multidimensional é através do Modelo Estrela, apresentado por Ralph Kimball (4). No centro da estrela encontra-se a tabela de fatos e, ao seu redor, as dimensões. Este modelo é apresentado na Figura 11:

**Figura 11: Modelo Estrela**

É um modelo simples e eficiente, caracterizado por possuir uma única tabela de fatos e chaves simples nas tabelas de dimensões. Cada dimensão é representada por uma única tabela.

Os pontos positivos deste modelo são a eficiência, dada pelo reduzido número de junções nas pesquisas e pelas chaves simples, e a facilidade de definir hierarquias.

Os pontos negativos são o tamanho e a desnormalização das tabelas de dimensões.

Modelo Estrela Parcial

É uma variação do Modelo Estrela, na qual existem várias tabelas fato e de dimensão separadas lógica e fisicamente por níveis de sumarização.

Desse modo, os dados são particionados em granularidades distintas. Por haver várias tabelas fato, na prática existem várias estrelas, cada uma representando uma combinação de níveis de agregação em cada dimensão. A Figura 12 apresenta uma parte do modelo que ilustra esta variação.

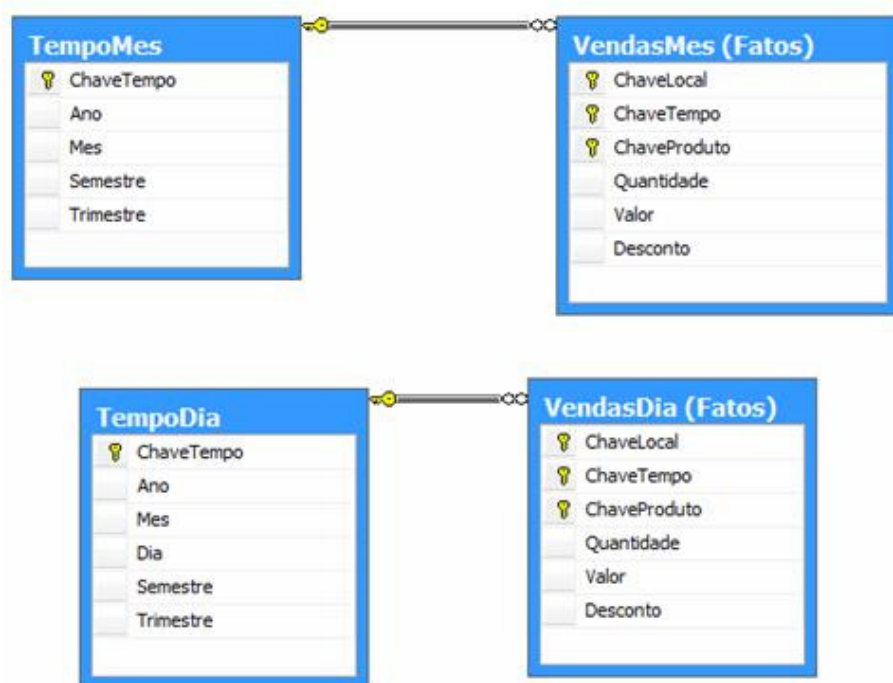


Figura 12: exemplo de duas estrelas no Modelo Estrela Parcial

Quando houver necessidade de novas agregações, basta criar outras tabelas com as granularidades desejadas, como ilustrado na Figura 13.

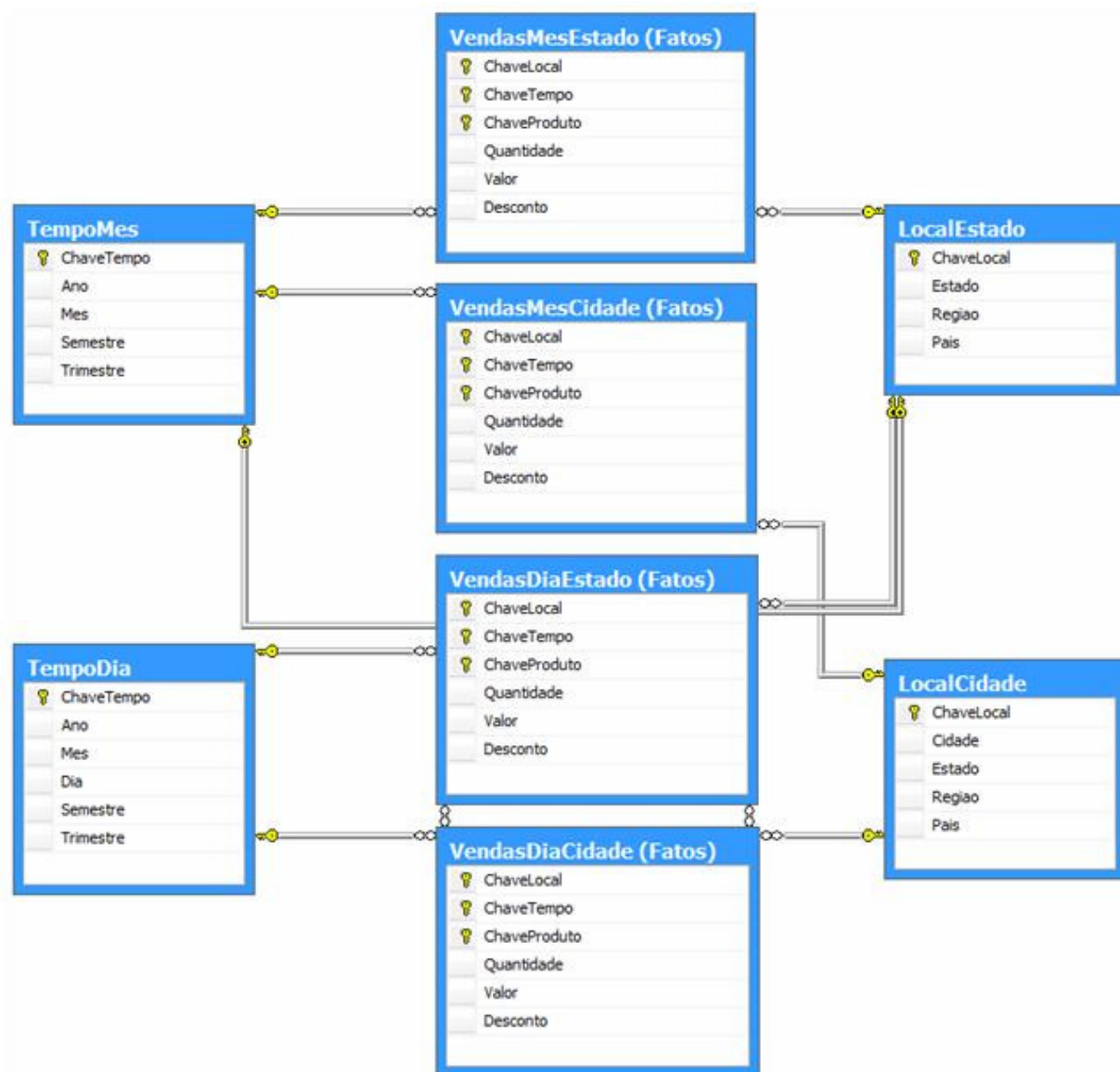


Figura 13: Modelo Estrela Parcial para composições de agregações

Os pontos positivos deste modelo são a maior economia de espaço, eliminando redundâncias e colunas que não têm sentido para determinado nível de agregação e o melhor desempenho para consultas de nível específico de agregação.

Por outro lado, a complexidade do modelo é maior e as consultas que combinam níveis de agregação distintos são mais elaboradas, podendo resultar em queda de desempenho.

Modelo Estrela com Particionamento de Fatos (ou Modelo Constelação de Fatos)

É uma variação do Modelo Estrela Parcial, na qual os fatos são particionados e as dimensões compartilhadas, conforme ilustrado na Figura 14.

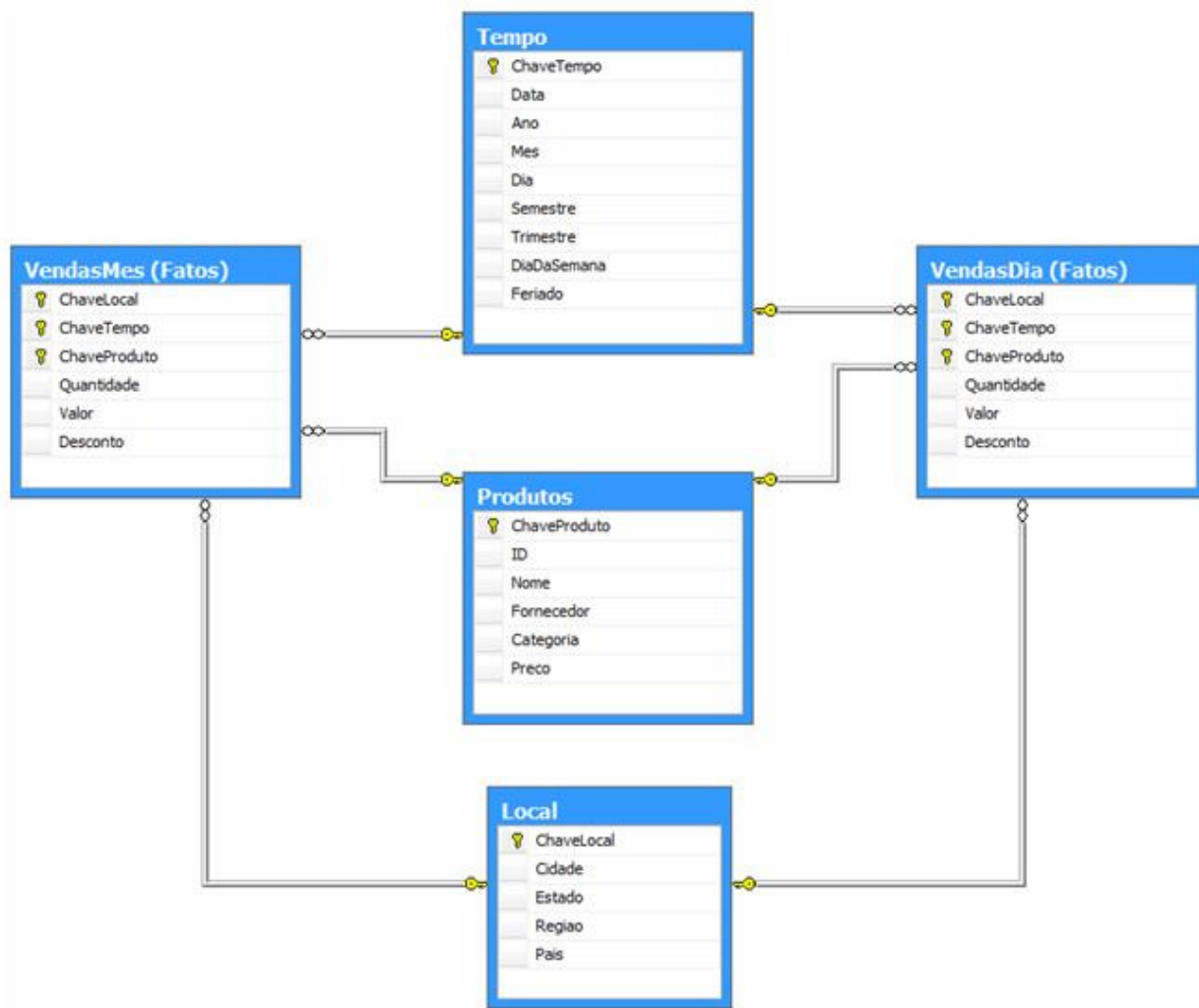


Figura 14: Modelo Particionamento de Fatos

Quando comparado ao Modelo Estrela Parcial, este modelo é menos exigente quanto à sua manutenção, dado o compartilhamento das tabelas de dimensão.

Modelo Estrela com Particionamento de Dimensões

Assim como o anterior, é uma variação do Modelo Estrela Parcial, porém com as dimensões particionadas, compartilhando a tabela de fatos. A Figura 15 apresenta este modelo. Note que a tabela de fatos deve conter os seus dados na maior granularidade que o modelo previr e também consolidados de acordo com os níveis mais altos.

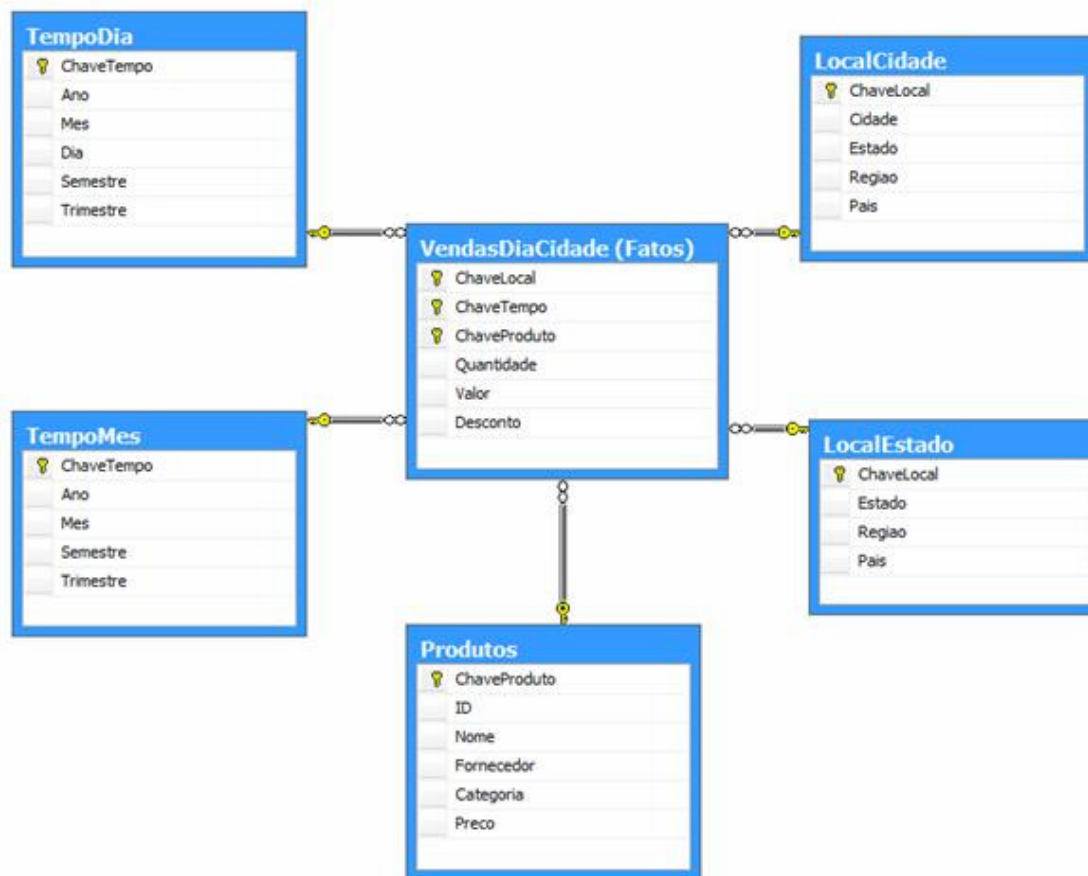


Figura 15: Modelo Particionamento de Dimensões, para local e tempo. Note a granularidade da tabela de fatos.

Este modelo é particularmente útil quando houver dimensões com grande quantidade de elementos, como é o caso de SKUs de produtos, por exemplo.

[↶ Início da página](#)

Snowflake e suas Variações

Os Modelos Snowflake acrescentam graus de normalização às tabelas de dimensões do Modelo Estrela, eliminando redundâncias e a necessidade do indicador NÍVEL. A Figura 16 mostra o resultado da normalização das tabelas “Produtos” e “Lojas” apresentadas na Figura 11. Observe a redução nas redundâncias, o que resulta em agilidade na manutenção. Apesar disso, é um modelo que resulta em maior número de tabelas em junções, podendo haver queda de desempenho.

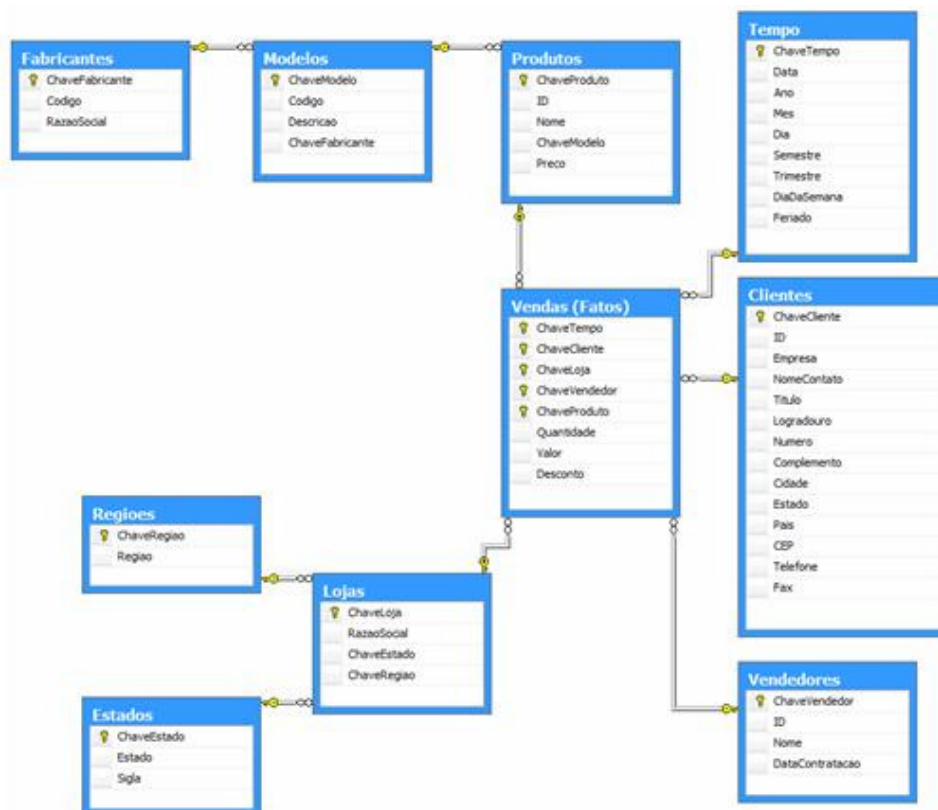


Figura 16: Modelo Snowflake, após normalização do Modelo Estrela da Figura 11

A seguir veremos três variações do Modelo Snowflake.

Modelo Snowflake Lookup

Neste modelo, ilustrado na Figura 17, as tabelas de dimensões são normalizadas, resultando na eliminação de redundâncias, o que torna a manutenção mais ágil e o modelo mais consistente.

Aqui, criamos uma tabela principal para uma determinada dimensão, que referencia tabelas de busca (lookup), estas contendo os nomes e descrições de campos.

Um cuidado extra com este modelo é o número de tabelas em junções, o que pode degradar o desempenho.

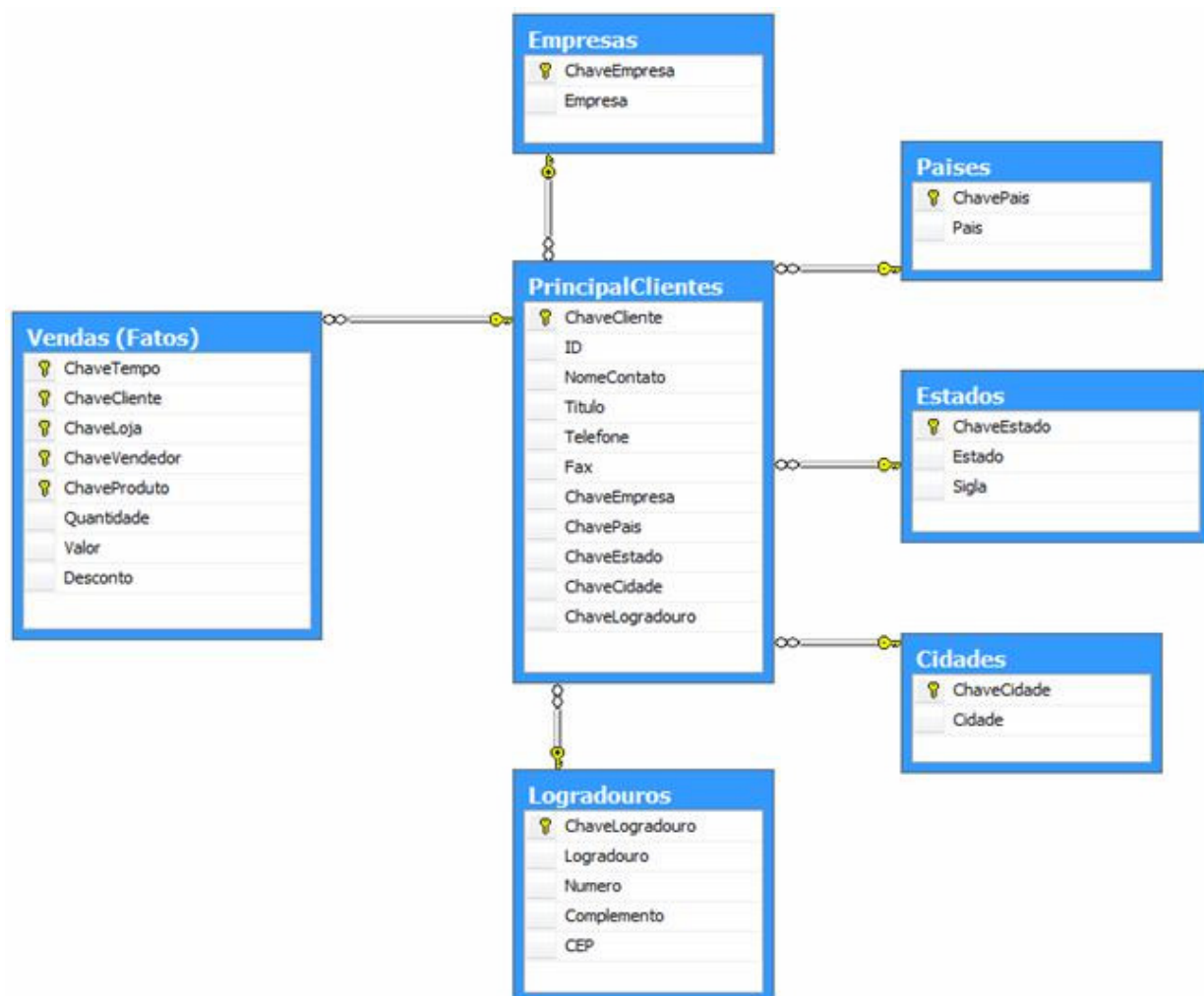


Figura 17: parte do Modelo Snowflake Lookup, mostrando a normalização da tabela Clientes do modelo da Figura 16

Observe no diagrama acima, que a tabela de fatos foi deslocada para a esquerda e nem todas as dimensões estão representadas, a fim de melhorar a visualização do modelo. Note que a tabela de dimensão "PrincipalClientes" possui apenas os dados de cada cliente e chaves estrangeiras para outros elementos, sendo que a manutenção destes é feita de modo mais consistente ao promover alterações apenas nas tabelas de busca (lookup).

Modelo Snowflake Chain

Este modelo encadeia as tabelas de dimensões começando com a tabela principal, que é o ponto de entrada para a tabela fato. A tabela principal da dimensão contém a chave para o próximo nível da hierarquia da dimensão e assim por diante. Na Figura 16, a normalização da dimensão "Produtos" em diversos níveis é um exemplo deste modelo. Note que a tabela de fatos possui indicação do nível mais baixo na hierarquia, referenciando a dimensão "Produtos", e então as tabelas de dimensões "Modelos" e "Fabricantes" percorrem os níveis mais altos.

A recomendação de uso deste modelo ocorre quando o nível de detalhe mais baixo está armazenado na tabela de fatos. A contra-indicação, por sua vez, é para os casos em que a pesquisa requer vários níveis de sumarização da informação, já que são necessários vários passos para recuperar as informações. A fim de melhorar o desempenho, uma sugestão é desnormalizar a cadeia, inserindo as chaves de níveis mais altos nos níveis mais baixos.

Modelo Snowflake Attribute

Com o objetivo de reduzir o número de informações referentes a atributos nas tabelas de fatos, geralmente utilizados para obtenção de detalhes (drillthrough), inserimos todos eles em uma tabela de atributos, conforme ilustrado pelas figuras a seguir.

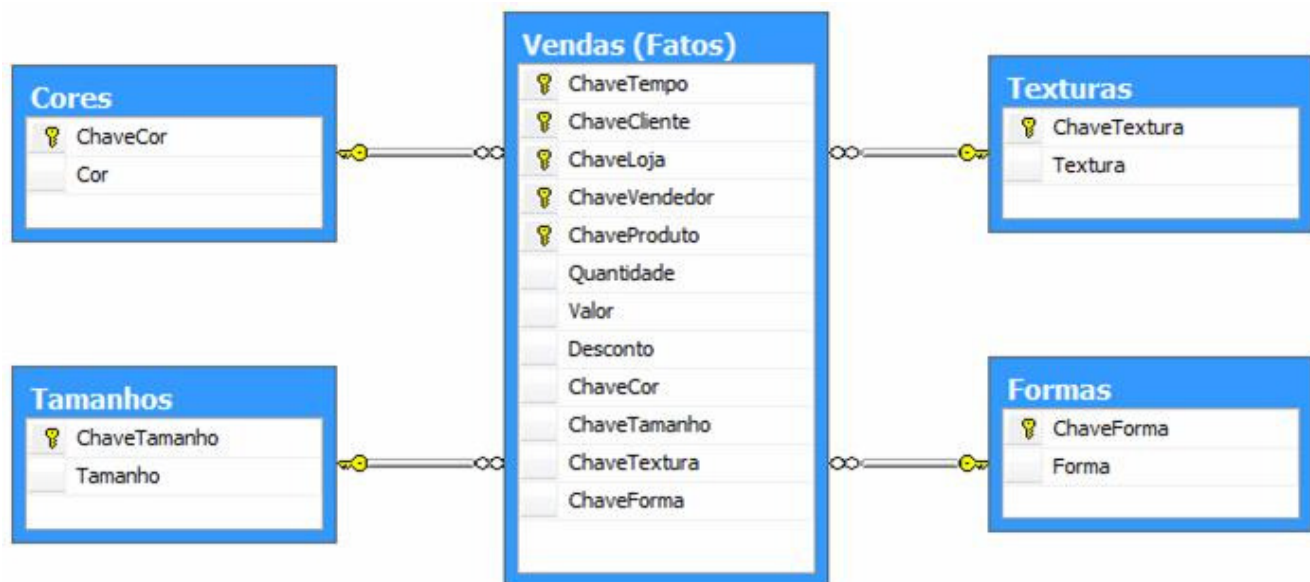


Figura 18: Modelo Snowflake, antes de separar os atributos

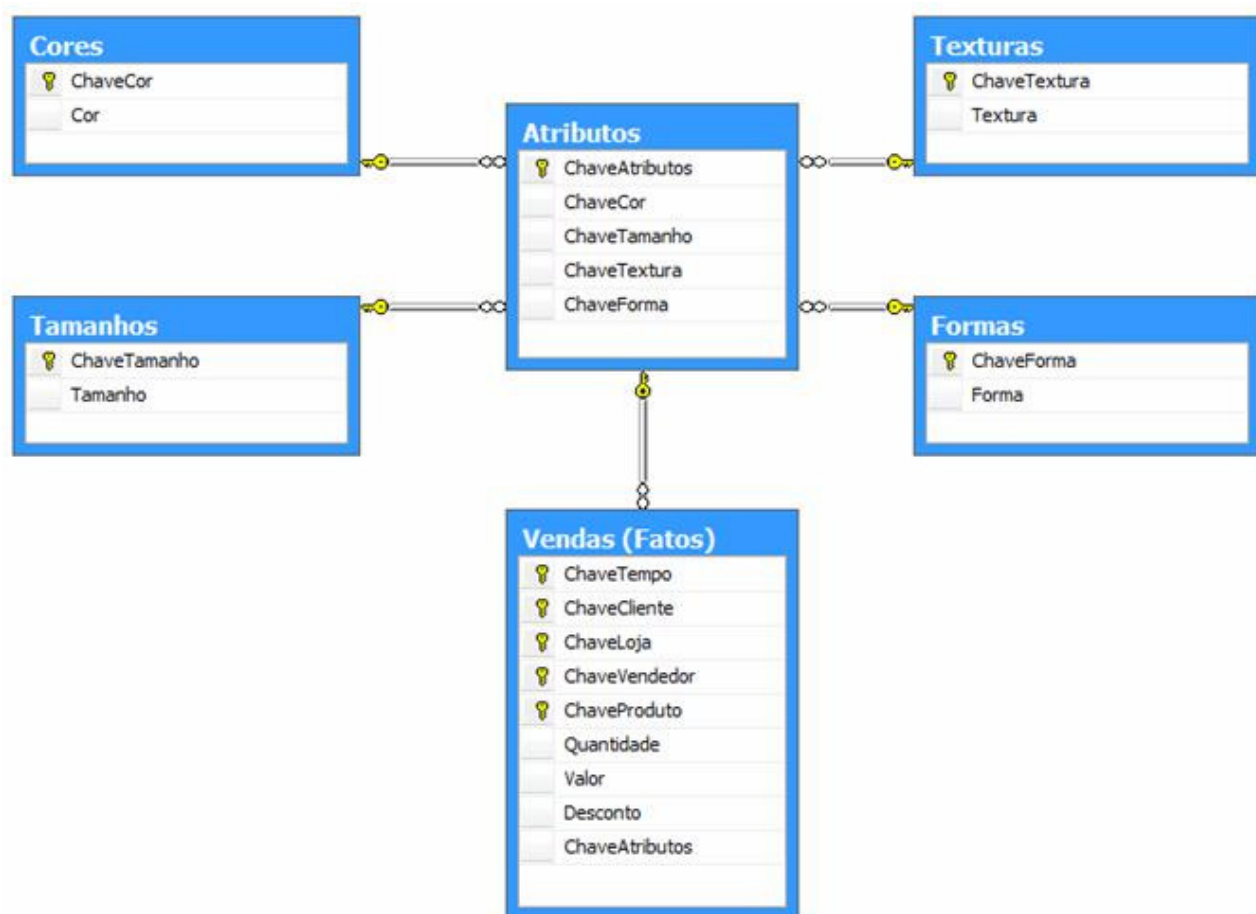


Figura 19: Modelo Snowflake Attribute

Outra utilidade deste modelo é a consolidação de informações sobre diversas pequenas dimensões que possuam poucos campos (muitas vezes apenas a descrição) em uma única tabela. Desse modo, o número de tabelas em junções pode ser reduzido, melhorando o desempenho.

[↩ Início da página](#)

Conclusão

O desenvolvimento de sistemas analíticos é cada vez mais comum. Embora haja ferramentas de diversos fornecedores, de nada elas adiantam se a modelagem de dados e o paradigma analítico não forem compreendidos.

Neste artigo procuramos percorrer assuntos pouco divulgados e com pouca bibliografia. Abordamos as estruturas de dados mais comumente encontradas em gerenciadores relacionais e analíticos: as B-Trees e as Árvore Bitmap. Ao conhecer um pouco da estrutura que suporta as tecnologias, esperamos que o leitor possa escolher melhor e entender as características

de cada modelo de dados que desenvolver e suportar.

As opções de modelagem são várias e aqui ilustramos algumas que podem ser utilizadas de modo isolado ou ainda combinadas, a fim de produzir modelos de dados multidimensionais que atendam a sua demanda.

[↶ Início da página](#)

Referências

1. PATRICIA – Practical Algorithm To Retrieve Information Coded In Alphanumeric. **Morrison, Donald R.** 4, 1968, JACM, Vol. 15, pp. 514-534.
2. **Knuth, Donald E.** The Art of Computer Programming. s.l. : Addison-Wesley, 1998. Vol. 3. ISBN 0-201-89685-0.
3. **Inmon, W H.** Building the Data Warehouse. s.l. : John Wiley & Sons, 1998.
4. **Ferreira, João Eduardo, Italiano, Isabel Cristina and Takai, Osvaldo Kotaro.** Introdução a Banco de Dados. [Online] 2005. [Cited: 06 21, 2007.] <http://www.ime.usp.br/~jef/apostila.pdf>.
5. **Tanler, Richard.** The Intranet Data Warehouse. s.l. : John Wiley & Sons, 1997.
6. **Kimball, Ralph.** The Data Warehouse Toolkit. s.l. : John Wiley & Sons, 2000.

[↶ Início da página](#)

[Fale Conosco](#)

©2007 Microsoft Corporation. Todos os direitos reservados. [Nota Legal](#) | [Marcas comerciais](#) | [Política de Privacidade](#)

Microsoft