



SISTEMA GERENCIADOR DE BANCO DE DADOS

AULA 4

Profª Vívian Ariane Barausse de Moura



CONVERSA INICIAL

O objetivo desta aula é apresentar os conceitos e definições sobre *data warehouse* (DW). Para isso, iniciaremos com os conceitos básicos, partiremos para a modelagem de dados com esse modelo e apresentaremos um projeto de *data warehouse*, com as questões pertinentes à construção de um DW. Também estudaremos sobre as ferramentas de DW e suas funcionalidades, e sobre as aplicações OLAP.

TEMA 1 – DATA WAREHOUSE

1.1 Conceitos básicos sobre *data warehouse*

Antes de iniciarmos as questões sobre o nosso tema, é importante relembrar algumas definições e terminologias, começando pelo conceito de *banco de dados*, que é definido por Elsmari e Navathe (2011, p. 720) “como uma coleção de dados relacionados”, e um *sistema de banco de dados* “é um banco de dados e um software de banco de dados juntos”.

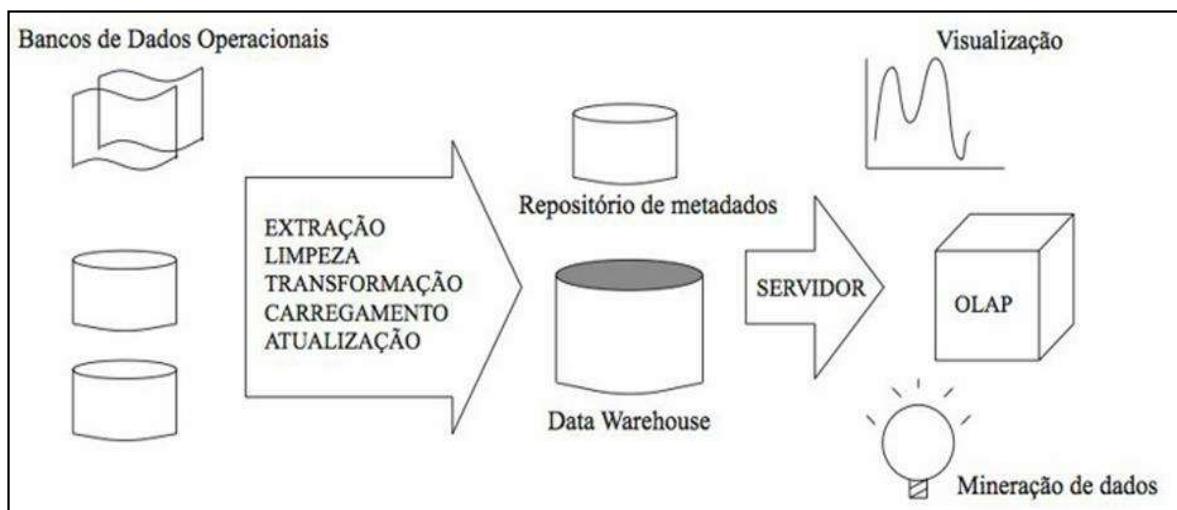
Os autores defendem que um *data warehouse* também é uma coleção de informações, assim como um sistema de suporte. A diferença é que os bancos de dados tradicionais são transacionais, podendo ser: relacionais, orientados a objetos, em rede ou hierárquicos. Um *data warehouse* tem a característica distintiva de servir principalmente para a aplicação de apoio à decisão, sendo utilizados para recuperar dados e não para processamento de transação de rotina, o que implica diretamente que os *data warehouse* são muito distintos dos bancos de dados tradicionais em sua estrutura, funcionamento, desempenho e finalidade.

Elsmari e Navathe (2011) afirmam que não existe uma única definição para *data warehouse*, pois eles têm sido desenvolvidos em organizações para atender a necessidades particulares. Os autores utilizam a definição de W. H. Inmon (Elsmari e Navathe, 2011, p. 720), que caracteriza um *data warehouse* “como uma coleção de dados orientada a assunto, integrada, não volátil, variável no tempo para o suporte às decisões da gerência”.

Os *data warehouses* oferecem acesso a dados para análise complexa, descoberta de conhecimento e tomada de decisão. Atualmente existe uma grande necessidade de oferecer aos que tomam decisões (da gerência

intermediária para cima) informações no nível correto de detalhe para dar suporte à atividade de tomada de decisão. Eles dão suporte a demandas de alto desempenho sobre os dados e informações de uma organização. O *data warehouse* em processamento analítico *on-line* OLAP e a mineração de dados oferecem essa funcionalidade (Elsmari; Navathe, 2011). A Figura 1 representa uma arquitetura típica de *data warehouse*.

Figura 1 – Arquitetura típica de *data warehouse*



Fonte: Ramakrishnan; Gehrke, 2008, p. 723.

Conforme já exposto, existem vários tipos de aplicações. Elmasri e Navathe (2011) apresentam as definições para OLAP, DSS e aplicações de mineração de dados, conforme o Quadro 1.

Quadro 1 – Definições

OLAP – Processamento analítico <i>on-line</i>	É um termo usado para descrever a análise de dados complexos do <i>data warehouse</i> . Nas mãos de trabalhadores do conhecimento habilidoso, as ferramentas OLAP utilizam capacidades de computação distribuída para análises que exigem mais armazenamento e poder de processamento do que pode estar localizada econômica e eficientemente em um <i>desktop</i> individual.
DSS sistema de apoio à decisão	Também conhecido como sistema de informações executivas. Não confunda com sistemas de integração empresarial; ajudam os principais tomadores de decisões de uma organização com dados de nível mais alto em decisões complexas e importantes.
Mineração de dados	É usada para descoberta do conhecimento – o processo de procurar novo conhecimento imprevisto nos dados. Estudaremos sobre este assunto em breve.

Fonte: Elsmari; Navathe, 2011, p. 721.



Ao fazer um comparativo sobre os suportes apresentados por bancos de dados tradicionais e *data warehouse*, temos as configurações apresentadas na Tabela 1.

Tabela 1 – Banco de dados tradicionais *versus* *data warehouse*

Banco de dados tradicionais	<i>Data warehouses</i>
O suporte para processamento de transação <i>on-line</i> (OLTP) que lida com inserções, atualizações e exclusões, enquanto também tem suporte para requisitos de consulta de informação. Os bancos de dados relacionais tradicionais são utilizados para processar consultas que podem tocar em uma pequena parte do banco de dados e transações que lidam com inserções ou atualizações no processo de algumas duplas por relação, assim eles não podem ser utilizados para OLPA, DSS ou mineração de dados.	São projetados exatamente para dar suporte à extração, processamento e apresentação eficiente para fins analíticos e de tomada de decisão. Em comparação com os bancos de dados tradicionais, os <i>data warehouses</i> em geral contêm quantidades muito grandes de dados de várias fontes, que podem incluir bancos de dados de diferentes modelos de dados, e às vezes arquivos adquiridos de sistemas e plataformas independentes.

Fonte: Elsmari; Navathe, 2011, p. 721.

De acordo com Elsmari e Navathe (2011), o crescente poder de processamento e a sofisticação das ferramentas e técnicas analíticas resultaram no desenvolvimento dos *data warehouses*, que oferecem armazenamento, funcionalidades e responsividade para as consultas além das capacidades dos bancos de dados orientados a transação. Acompanhando esse poder cada vez maior está uma grande demanda para melhorar o desempenho de acesso aos dados dos bancos de dados

TEMA 2 – MODELAGEM DE DADOS PARA DATA WAREHOUSE

Para modelagem de dados, de acordo com Elsmari e Navathe (2011), deve ser levada em consideração a tipologia. Em modelos multidimensionais ocorre o relacionamento dos dados em matrizes multidimensionais, que são chamadas de *cubo de dados*. Também pode ocorrer de haver mais de três dimensões e, nesse caso, a denominação é *hipercubo*. Se compararmos ao modelo de dados relacional, o desempenho na consulta das matrizes multidimensionais pode ser muito melhor nos dados que estão na formação dimensional. Os autores utilizam três exemplos de dimensões em um *data warehouse* corporativo, que são: os períodos fiscais, produtos e regiões da empresa.

Utilizando como exemplo uma planilha de vendas regionais por produto para determinado período, os produtos poderiam ser mostrados como linhas,



com as receitas de venda para cada região compreendendo as colunas. A Figura 2 mostra essa organização, que é uma planilha padrão, sendo matriz bidimensional (Elsmari; Navathe, 2011).

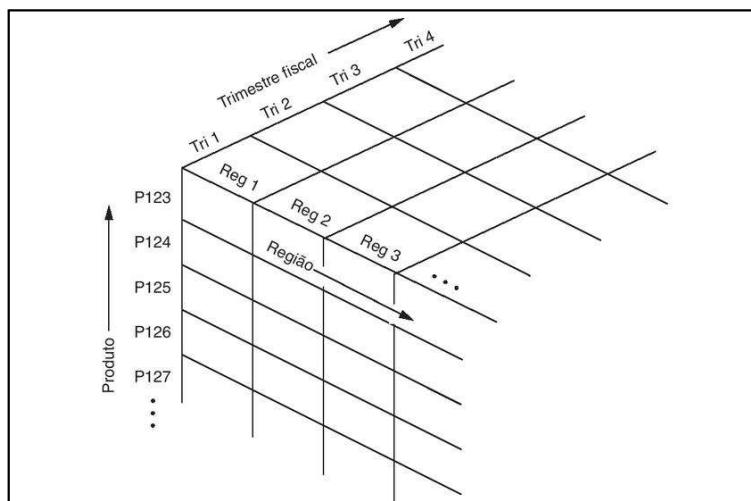
Figura 2 – Um modelo de matriz bidimensional

		Região			
		Reg 1	Reg 2	Reg 3	...
Produto	P123				
	P124				
	P125				
	P126				
	:				

Fonte: Elsmari; Navathe, 2011, p. 723.

Ao acrescentar uma dimensão de tempo, como os trimestres fiscais de uma organização, seria produzida uma matriz tridimensional, que poderia ser representada usando um cubo de dados, conforme se vê na Figura 3.

Figura 3 – Um modelo de cubo de dados tridimensional



Fonte: Elsmari; Navathe, 2011, p. 723.

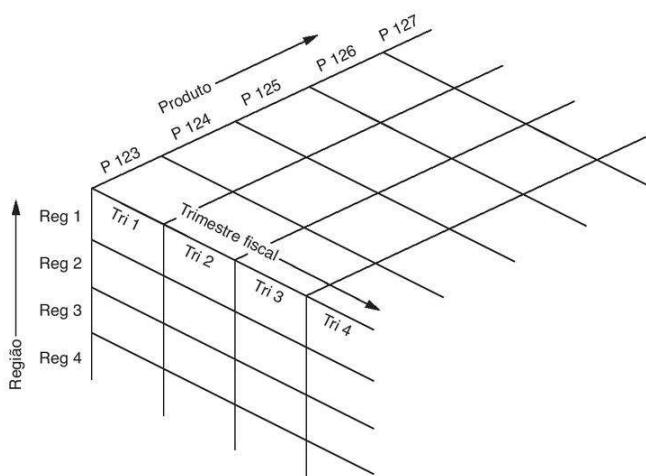
Na Figura 3, Elsmari e Navathe (2011) utilizam um cubo de dados tridimensional que organiza os dados de vendas de produtos por trimestres fiscais e regiões de venda. Cada célula teria dados para um produto específico, trimestre fiscal específico e região específica. Ao incluir outras dimensões, o hipercubo de dados poderia ser produzido, embora se houver mais de três dimensões não poderão ser facilmente visualizadas ou apresentadas de maneira



gráfica. Os dados podem ser consultados diretamente em qualquer combinação de dimensões, evitando consultas de banco de dados complexas. Existem ferramentas para visualizar dados de acordo com a escolha de dimensões do usuário.

A mudança de hierarquia, de acordo com Elsmari e Navathe (2011, p. 722), de uma orientação unidimensional para outra é algo feito com facilidade em um cubo de dados com uma técnica chamada de *giro*, que também pode ser chamada de *rotação*. Nessa técnica, o cubo de dados pode ser imaginado girando para mostrar uma orientação diferente dos eixos. Por exemplo: imagine o giro do cubo de dados, para mostrar as receitas de vendas regionais como linhas, os totais de receita por trimestre fiscal como colunas e os produtos da empresa na terceira dimensão, conforme representado na Figura 4.

Figura 4 – Versão girada do cubo de dados da Figura 3



Fonte: Elsmari; Navathe, 2011, p. 724.

De acordo com os autores, essa técnica é equivalente a ter uma tabela de vendas regionais para cada produto separadamente, em que cada tabela mostra vendas trimestrais para esse produto região por região (Elsmari; Navathe, 2011).

Elsmari e Navathe (2011, p. 723) explicam que modelos multidimensionais atendem prontamente a visões hierárquicas, conhecidas como exibição *roll-up* ou exibição *drill-down*. Uma exibição *roll-up* sobe na hierarquia, agrupando em unidades maiores ao longo de uma dimensão. Por exemplo, somando dados semanais por trimestre ou por ano. A Figura 5 mostra uma exibição *roll-up*, que move de produtos individuais para uma categorização maior dos produtos.



Figura 5 – A operação *roll-up*

Categorias de produtos	Região		
	Região 1	Região 2	Região 3
Produtos 1XX			
Produtos 2XX			
Produtos 3XX			
Produtos 4XX			

Fonte: Elsmari; Navathe, 2011, p. 724.

Uma exibição *drill-down* oferece a capacidade oposta, fornecendo uma visão mais detalhada. Talvez desagregando as vendas do país por região e, depois, as vendas regionais por sub-região e também separando produtos por estilos (Elsmari; Navathe, 2011, p. 724), conforme vemos na Figura 6.

Figura 6 – A operação *drill-down*

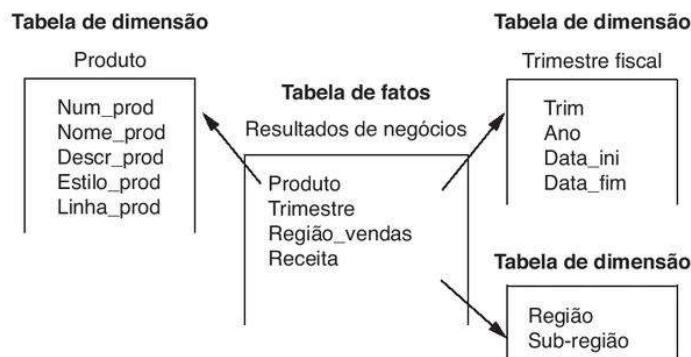
		Região 1		Região 2	
		Sub_reg 1	Sub_reg 2	Sub_reg 3	Sub_reg 4
Estilos P123	A				
	B				
	C				
	D				
Estilos P124	A				
	B				
	C				
Estilos P125	A				
	B				
	C				
	D				

Fonte: Elsmari; Navathe, 2011, p. 725.

Segundo Elsmari e Navathe (2011, p. 724), o modelo de armazenamento multidimensional envolve dois tipos de tabelas: a *tabela de dimensão*, que consiste em tuplas de atributos da dimensão, e a *tabela de fatos*, que pode ser imaginada como sendo tuplas, uma para cada fato registrado. Esse fato contém alguma variável observada e a identifica com ponteiros para tabelas de dimensão, podendo ser uma ou várias variáveis. A tabela de fatos contém os dados, e as dimensões identificam cada tupla nesses dados. A Figura 7 contém um exemplo de tabela de fatos que pode ser vista do ponto de vista de múltiplas tabelas de dimensão.



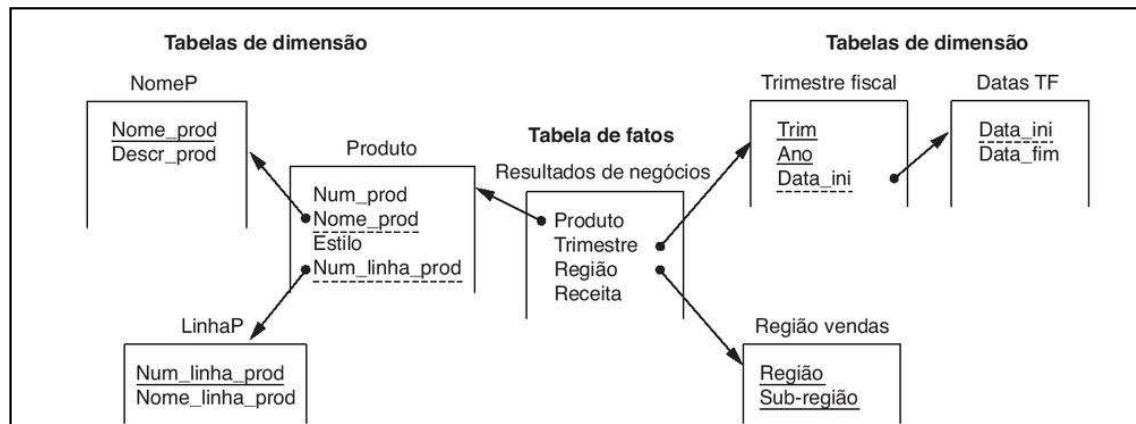
Figura 7 – Um esquema de estrela com tabelas de fato e dimensões



Fonte: Elsmari; Navathe, 2011, p. 725.

Dois esquemas multidimensionais comuns são o esquema estrela e o esquema floco de neve. O *esquema estrela* consiste em uma tabela de fatos com uma única tabela para cada dimensão, conforme a Figura 7. O *esquema floco de neve* é uma variação do esquema estrela, em que as tabelas dimensões de um esquema estrela são organizadas em uma hierarquia ao normalizá-las, como pode ser visto na Figura 8.

Figura 8 – Um esquema floco de neve

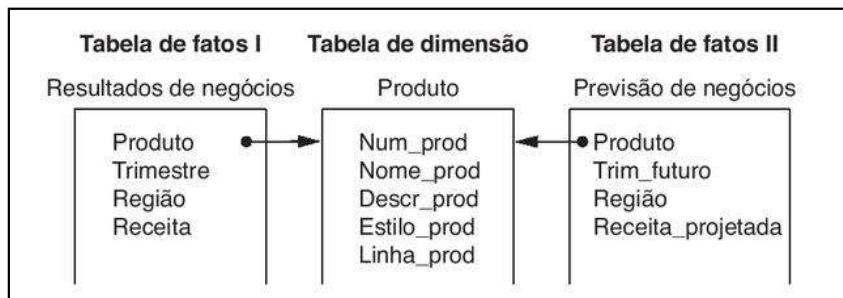


Fonte: Elsmari; Navathe, 2011, p. 726.

Elsmari e Navathe (2011, p. 725) defendem que algumas instalações estão normalizando *data warehouses* até a terceira forma normal, de modo que possam acessá-los até o maior nível de detalhe. Uma *constelação de fatos* é um conjunto de tabela de fatos que compartilham algumas tabelas de dimensão. A Figura 9 mostra uma constelação de fatos com duas tabelas de fatos, regras de negócios e previsão de negócios, as quais compartilham a tabela de dimensão chamada *produto*. As constelações de fatos limitam as possíveis consultas para o *data warehouse*.



Figura 9 – Uma constelação de fatos



Fonte: Elsmari; Navathe, 2011, p. 726.

TEMA 3 – PROJETO DE DATA WAREHOUSE

Para a construção de um *data warehouse*, segundo Elsmari e Navathe (2011), os responsáveis devem ter uma visão ampla do uso antecipada, pois não existe um meio de antecipar todas as consultas ou análises possíveis durante a fase do projeto, porém os mesmos autores salientam que o projeto deve aceitar especificamente a consulta ocasional, ou seja, é possível acessar dados com qualquer combinação significativa de valores para os atributos nas tabelas de dimensão ou fatos. Por exemplo, uma empresa de produtos de consumidor com *marketing* intenso exigiria diferentes maneiras de organizar o *data warehouse* em relação a uma empresa de caridade sem fins lucrativos voltada para angariar fundos. Um esquema apropriado seria escolhido para refletir o uso antecipado. Os autores elaboraram algumas etapas no que diz respeito à aquisição de dados para o *data warehouse* que estão dispostas no Quadro 2.

Quadro 2 – Etapas de aquisição de dados de um *data warehouse*

1	Os dados precisam ser extraídos de várias fontes heterogêneas. Por exemplo, banco de dados ou outras entradas de dados, como aquelas que contêm dados do mercado financeiro ou dados ambientais.
2	Os dados precisam ser formatados por coerência dentro do <i>data warehouse</i> . Nomes, significados e domínios de dados de fontes não relacionadas precisam ser reconciliados. Por exemplo, empresas subsidiárias de grandes corporações podem ter diferentes calendários fiscais com trimestres terminados em datas diferentes, tornando difícil agregar dados financeiros por trimestre.

(continua)

3	<p>Os dados precisam ser limpos para garantir a validade. A limpeza de dados é um processo complicado e complexo, que tem sido identificado como componente que mais exige trabalho na construção do <i>data warehouse</i>. A entrada de dados precisa ser examinada e formatada de modo consistente, sendo verificada a validade e a qualidade. Reconhecer dados errôneos e incompletos é difícil de automatizar, e a limpeza que requer correção de erro automática pode ser ainda mais complicada. Por exemplo, pode-se exigir que "Cidade=Campinas" junto com o "Estado=RJ" seja reconhecido como uma combinação incorreta. Depois que tais problemas tiverem sido resolvidos, dados semelhantes de fontes diferentes precisam ser coordenados para carregar no <i>data warehouse</i>. O processo de retornar dados limpos para origem é chamado de <i>fluxo reverso</i>.</p>
4	<p>Os dados precisam ser ajustados ao modelo de dados do armazém. Baixar os dados de várias fontes que devem ser instalados no modelo de dados no <i>data warehouse</i>. Eles podem ter que ser convertidos de banco de dados relacionais, orientados a objetos ou legados (em rede e/ou hierárquico) para um modelo multidimensional.</p>
5	<p>Os dados precisam ser carregados no <i>data warehouse</i>. O grande volume de dados torna a carga dos dados uma tarefa significativa; são necessárias ferramentas de monitoramento para cargas, bem como métodos para recuperação de cargas incompletas ou incorretas. Com o imenso volume de dados no <i>data warehouse</i>, a atualização incremental normalmente é a única técnica viável.</p>

Fonte: Elsmari; Navathe, 2011, p. 727.

Conforme Elsmari e Navathe (2011, p. 727), os bancos de dados precisam lutar por um equilíbrio entre eficiência no processamento de transação e suporte dos requisitos da consulta, mas um *data warehouse* normalmente é utilizado para acesso com base nas necessidades de um tomador de decisão. Nesse sentido, o armazém de dados em um *data warehouse* reflete essa especialização e envolve os processos descritos na Tabela 2.

Tabela 2 – Processos armazenamento de dados

- Armazenamento dos dados de acordo com o modelo de dados do armazém;
- Criação e manutenção das estruturas de dados exigidas;
- Criação e manutenção dos caminhos de acesso apropriados;
- Fornecimento de dados variáveis no tempo à medida que novos dados são incluídos;
- Suporte e atualização dos dados do *data warehouse*;
- Atualização dos dados;
- Eliminação dos dados.

Fonte: Elsmari; Navathe, 2011, p. 727.

Os autores destacam que o imenso volume de dados torna impossível recarregá-los em sua totalidade. As alternativas são a atualização seletiva dos dados e versões, e *data warehouses* separadas. Quando *data warehouses* utilizam o mecanismo de atualização de dados incremental, os dados precisam ser periodicamente eliminados. Por exemplo: um armazém que mantém dados



sobre os 12 trimestres comerciais anteriores pode, de maneira periódica, eliminar seus dados a cada ano (Elsmari; Navathe, 2011).

Os *data warehouses* também devem ser projetados com consideração total do ambiente em que residiram, em que é importante incluir os elementos disponíveis na Tabela 3.

Tabela 3 – Elementos para projeto do *data warehouse*

- Projeções de uso;
- O ajuste de modelo de dados;
- Características das fontes disponíveis;
- Projeto do componente de metadados;
- Projeto de componente modular;
- Projeto de facilidade de gerenciamento de mudança;
- Considerações de arquitetura distribuída e paralela.

Fonte: Elsmari; Navathe, 2011, p. 728.

Conforme já especificados por Elsmari e Navathe (2011), o projeto de *data warehouse* é inicialmente controlado por projeção de uso, ou seja, por expectativas sobre quem usará e como usarão. A escolha de um modelo de dados para dar suporte a esse uso é uma decisão inicial chave, e as projeções de uso e as características das origens dos dados do *data warehouse* devem ser levadas em consideração. Um *data warehouse* bem montado deve ser projetado para facilidade de manutenção, permitindo que os gerentes de *data warehouse* planejem e gerenciem a mudança com eficiência enquanto oferecem suporte ideal para os usuários.

A arquitetura do ambiente de computação distribuída da organização é uma importante característica determinante para o projeto de *data warehouse*. Existem duas arquiteturas distribuídas básicas: o *data warehouse* distribuídos e o *data warehouse* federado, cujas características estão presentes no quadro 3.



Quadro 3 – *Data warehouse* distribuído e federado

<i>Data warehouse</i> distribuído	<i>Data warehouse</i> federado
Todos os aspectos dos bancos de dados distribuídos são relevantes, por exemplo: replicação, particionamento, comunicação e questões de consistência. Uma arquitetura distribuída pode oferecer benefícios particularmente importantes ao desempenho do armazém, como balanceamento de carga melhorado, estabilidade de desempenho e maior disponibilidade. Um único repositório de metadados replicados residiria em cada site de distribuição.	Tem a mesma ideia do banco de dados federado, em que uma confederação descentralizada de <i>data warehouses</i> autônomos, cada um com o próprio repositório de metadados. Dada a magnitude do desafio inerente aos <i>data warehouses</i> , é provável que tais federações consistam em componentes escala menor como os <i>data marts</i> . Grandes organizações podem decidir considerar <i>data marts</i> em vez de montar <i>data warehouses</i> imensos.

Fonte: Adaptada de Elsmari e Navathe, 2011, p. 728.

TEMA 4 – FERRAMENTAS DE DATA WAREHOUSE (DATA MART)

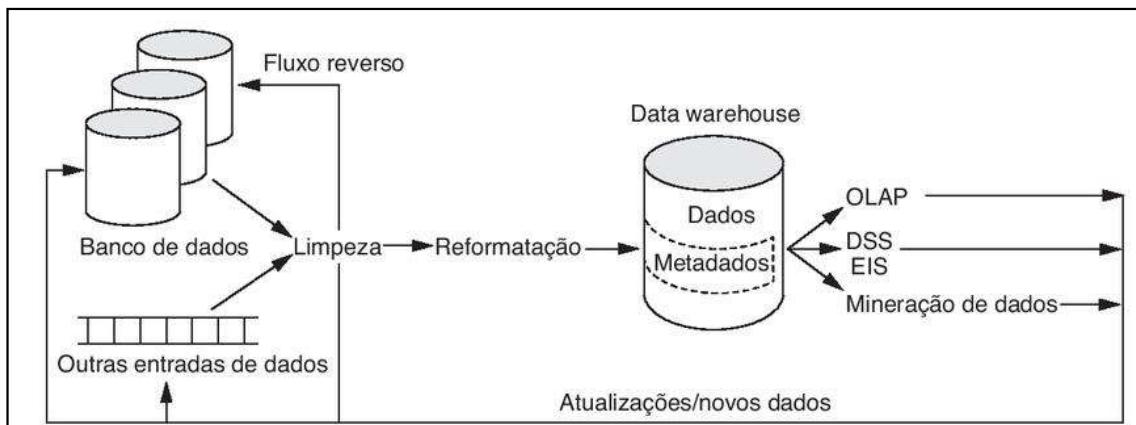
Segundo Elsmari e Navathe (2011, p. 720), o modelo de dados multidimensional é uma boa escolha para OLAP e tecnologias de apoio à decisão. Um *data warehouse* com frequência é um depósito de dados integrados de múltiplas fontes, processados para armazenamento em um modelo multidimensional e é diferente da maioria dos bancos de dados transacionais. Os *data warehouses* costumam apoiar a análise de série temporal e tendência, ambas exigindo mais dados históricos do que geralmente é mantido nos bancos de dados transacionais.

As informações nos *data warehouses*, conforme Elsmari e Navathe (2011), mudam com muito menos frequência e podem ser consideradas não de tempo real, com atualização periódica. A informação é muito menos detalhada e atualizada de acordo com uma escolha cuidadosa de política de atualizações, normalmente incremental. As atualizações no armazém são tratadas pelo componente de aquisição do armazém que oferece todo o pré-processamento exigido.

Os autores descrevem o *data warehouse* como “uma coleção de tecnologias de apoio à decisão, visando habilitar o trabalhador do conhecimento (executivo, gerente, analista) a tomar decisões melhores e mais rápidas”. (Elsmari; Navathe, 2011, p. 721). A Figura 10 oferece uma visão geral da estrutura conceitual de um *data warehouse*.



Figura 10 – Exemplo de transações no modelo de cesta de mercado



Fonte: Elsmari; Navathe, 2011, p. 722.

A Figura 10 mostra o processo inteiro que inclui a possível limpeza e reformatação dos dados antes que sejam carregados no armazém. Esse processo é tratado por ferramentas conhecidas como ETL – extração, transformação e carga. No *backend* do processo, OLAP, mineração de dados e DSS podem gerar novas informações relevantes, como as regras. Essas informações aparecem na figura voltando ao armazém. A figura também mostra que as fontes de dados podem incluir arquivos. As características dos *data warehouses* estão presentes na Tabela 4.

Tabela 4 – Características diferenciadoras dos *data warehouse*

- Visão conceitual multidimensional;
- Dimensionalidade genérica;
- Dimensões e níveis de agregação ilimitados;
- Operações e restritas entre dimensões;
- Tratamento dinâmico de Matriz esparsa;
- Arquitetura cliente-servidor;
- Suporte para múltiplos usuários;
- Arquitetura cliente-servidor;
- Acessibilidade;
- Transparência;
- Manipulação de dados intuitiva;
- Desempenho de relatório consistente;
- Recurso de relatório flexível.

Fonte: Elsmari; Navathe, 2011, p. 721.

De acordo com Elsmari e Navathe (2011), como os *data warehouses* abrangem um grande volume de dados, geralmente são uma ordem de magnitude maior que o banco de dado de origem. O imenso volume de dados, em geral na faixa dos *terabyte* ou *petabytes*, é uma questão que tem sido tratada



por meio de *data warehouses* em nível empresarial, *data warehouses* virtuais e *data marts* (Quadro 4).

Quadro 4 – *Data warehouse*

Data warehouses em nível empresarial	São imensos projetos que exigem investimento maciço de tempo e recursos.
Data warehouses virtuais	Oferecem visões de banco de dados operacionais que são materializadas para acesso eficiente
Data marts	Em geral são voltados para um subconjunto da organização como um departamento e possuem um foco mais estreito.

Fonte: Elsmari; Navathe, 2011, p. 722.

Elsmari e Navathe (2011) defendem que os *data warehouses* existem para facilitar as consultas ocasionais complexas. Com o uso intenso de dados frequentes, precisam oferecer suporte à consulta muito maior e mais eficiente do que é exigido dos bancos de dados transacionais. O componente de acesso ao *data warehouse* tem suporte para funcionalidade de planilha avançada, processamento de consulta eficiente, consultas estruturadas, consultas ocasionais, mineração de dados e visões materializadas. Em particular, a funcionalidade de planilha avançada inclui suporte para as mais modernas aplicações de planilha, por exemplo, MS-Excel, bem como para programas de aplicações OLAP. Estes oferecem funcionalidades pré-programadas apresentadas no Quadro 5.

Quadro 5 – Funcionalidades pré-programadas

Roll-up	Os dados são resumidos com generalização cada vez maior, por exemplo: semanal para trimestral para anual.
Drill-down	Níveis cada vez maiores de detalhes são revelados (complemento de <i>roll-up</i>).
Giro	É realizada a tabulação cruzada, também conhecida como <i>rotação</i> .
Slice e dice	Operações de projeção são realizadas nas dimensões.
Ordenação	Os dados são ordenados por valor ordinal.
Seleção	Os dados estão disponíveis por valor ou intervalo.
Atributos derivados (calculados)	Atributos são calculados por operações sobre valores armazenados e derivados.

Fonte: Elsmari; Navathe, 2011, p. 728.

Elsmari e Navathe (2011) afirmam que, como os *data warehouses* são livres das restrições do ambiente transacional, existe uma eficiência aumentada no processamento da consulta. Entre as ferramentas e técnicas usadas estão a transformação de consulta; interseção e união de índice; funções especiais



ROLAP (OLAP relacional) e MOLAB (OLAP multidimensional); extensões SQL; métodos de junção avançados; e varredura inteligente, como no acréscimo de consultas múltiplas.

Segundo Elsmari e Navathe (2011), o melhor desempenho também tem sido obtido com o processamento paralelo. As arquiteturas de servidor paralelas incluem multiprocessador simétrico (SMP), *cluster* e processamento maciçamente paralelo (MPP), além de combinações destes.

Os trabalhadores do conhecimento e os tomadores de decisão utilizam ferramentas que variam de consultas parametrizadas até consultas ocasionais e mineração de dados. Assim, o componente de acesso do *data warehouse* precisa oferecer suporte para consultas estruturadas, tanto parametrizadas como ocasionais, pois juntos eles compõem o ambiente de consulta gerenciado.

TEMA 5 – SERVIDORES OLAP

De acordo com Ramakrishnan e Gehrke (2008, p. 704), “as aplicações de OLAP são dominadas por consultas *ad hoc* complexas. Em termos de SQL, essas consultas envolvem operadores de agrupamento e agregação”. O autor salienta que a maneira natural de pensar sobre consultas OLAP típicas é em termos de um modelo de dados multidimensional. No modelo de dados multidimensional, o foco é em uma coleção de medidas numéricas, em que cada medida depende de um conjunto de dimensões.

O autor utiliza um exemplo baseado em dados de vendas. O atributo de medida do exemplo é *vendas*. As dimensões são *produto*, *local* e *tempo*. Dado um produto, um local e um tempo, temos no máximo um valor de vendas associado. Se identificamos um produto por um identificador único *idp* e, analogamente, identificamos o local por *idloc* e o tempo por *idtempo*, podemos considerar as informações de vendas como organizadas em um array tridimensional Vendas. Esse array aparece na Figura 11, mostrando apenas os valores de um único *idloc*, *idloc=1*, que pode ser considerado um corte ortogonal no eixo *idloc*.



Figura 11 – Vendas: um conjunto de dados multidimensional

idloc		
13	8	10
12	30	20
11	25	8
	1	2
		3

Fonte: Ramakrishnan; Gehrke, 2008, p. 706.

Conforme Ramakrishnan e Gehrke (2008), essa visão dos dados como um *array* multidimensional é facilmente generalizada para mais de três dimensões. Em aplicações OLAP, a maior parte dos dados pode ser representada em tal *array* multidimensional. Na verdade, alguns sistemas OLAP realmente armazenam dados em um *array* multidimensional (naturalmente, implementado sem a suposição usual de linguagem de programação de que o *array* inteiro cabe na memória). Os sistemas OLAP que usam *arrays* para armazenar conjuntos de dados multidimensionais são chamados de sistemas OLAP multidimensionais (MOLAP – multidimensional OLAP).

Ramakrishnan e Gehrke (2008) afirmam que, em um *array* multidimensional, os dados também podem ser representados como uma relação, conforme ilustrado na Figura 12, que mostra os mesmos dados da Figura 11, com linhas adicionais correspondendo ao “corte” $idloc=2$. Essa relação, que relaciona as dimensões com a medida de interesse, é chamada de *tabela de fatos*.



Figura 12 – Locais, produtos e vendas representados como relações

<i>idloc</i>	<i>cidade</i>	<i>estado</i>	<i>país</i>
1	Madison	WI	EUA
2	Fresno	CA	EUA
5	Chennai	TN	Índia
Locais			
<i>idp</i>	<i>nomep</i>	<i>categoria</i>	<i>preço</i>
11	Jeans Lee	Vestuário	25
12	Zord	Brinquedos	18
13	Caneta Biro	Artigos de escritório	2
Produtos			
<i>idp</i>	<i>idtempo</i>	<i>idloc</i>	<i>vendas</i>
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26
12	2	2	45
12	3	2	20
13	1	2	20
13	2	2	40
13	3	2	5
Vendas			

Fonte: Ramakrishnan; Gehrke, 2008, p. 707.

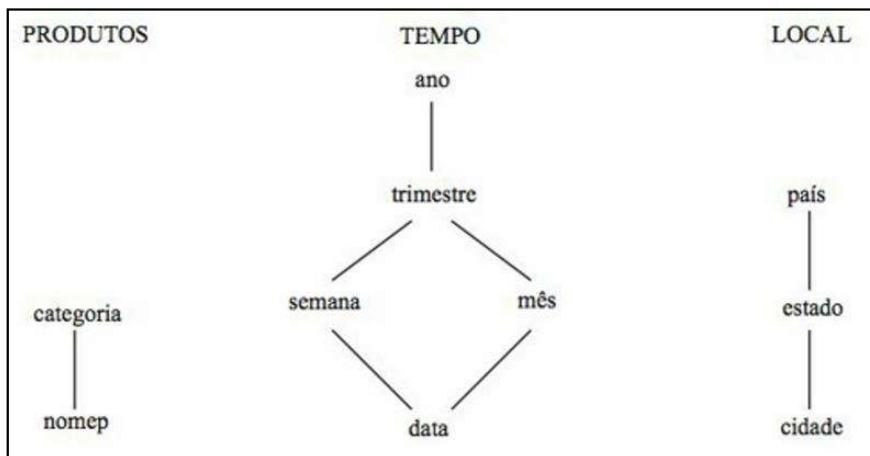
Em relação às dimensões, Ramakrishnan e Gehrke (2008) defendem que cada dimensão pode ter um conjunto de atributos associados. Por exemplo, a dimensão local é identificada pelo atributo *idloc*, que usamos para identificar um local na tabela *Vendas*. Supomos que ela também tenha os atributos *país*, *estado* e *cidade*. Supomos ainda que a dimensão *Produto* tenha os atributos *nomep*, *categoria* e *preço*, além do identificador *idp*. A *categoria* de um produto indica sua natureza geral (por exemplo, um produto calça poderia ter um valor de categoria *vestuário*). Supomos que a dimensão *Tempo* tenha os atributos *data*, *semana*, *mês*, *trimestre*, *ano* e *flag_feriado*, além do identificador *idtempo*.

Ramakrishnan e Gehrke (2008) expõem que, para cada dimensão, o conjunto de valores associados pode ser estruturado como uma hierarquia. Por exemplo, as cidades pertencem aos estados, e os estados pertencem aos países. As datas pertencem às semanas e aos meses; as semanas e os meses estão contidos em trimestres, e os trimestres estão contidos nos anos (note que uma semana poderia abranger dois meses; portanto, as semanas não estão contidas nos meses). Alguns dos atributos de uma dimensão descrevem a



posição de um valor de dimensão com relação a essa hierarquia de valores de dimensão subjacente. As hierarquias de *produto*, *local* e *tempo* de nosso exemplo aparecem no nível de atributo na Figura 13.

Figura 13 – Hierarquias de dimensão



Fonte: Ramakrishan; Gehrke, 2008, p. 708.

As informações sobre dimensões também podem ser representadas como uma coleção de relações:

Figura 14 – Relações

```
Locais(idloc: integer, cidade: string, estado: string, país: string)
Produtos(idp: integer, nomep: string, categoria: string, preço: real)
Tempos(idtempo: integer, data: string, semana: integer, mês: integer,
       trimestre: integer, ano: integer, flag_feriado: boolean)
```

Fonte: Ramakrishan; Gehrke, 2008, p. 708.

Essas relações são muito menores do que a tabela de fatos em uma aplicação OLAP típica; elas são chamadas de *tabelas de dimensão*. Os sistemas OLAP que armazenam todas as informações, incluindo as tabelas de fatos, como relações, são chamados de *sistemas OLAP relacionais* (ROLAP – relational OLAP) A tabela *Tempos* ilustra a atenção prestada à dimensão tempo nas aplicações OLAP típicas. Os tipos de dados *date* e *timestamp* da SQL não são adequados; para suportar resumos que refletem operações comerciais, são mantidas informações como trimestres fiscais, feriados etc., para cada valor de tempo.



FINALIZANDO

Esta aula focou no estudo do *data warehouse*, que pode ser visto como um processo que requer uma série de atividades preliminares, ao contrário da mineração de dados, que pode ser imaginada como uma atividade que retira conhecimento de um *data warehouse*. Neste sentido, estudamos os principais conceitos relacionados ao *data warehouse*. Foram apresentadas as modelagens de dados e relacionadas algumas etapas sobre o projeto de *data warehouse*, assim como as questões pertinentes às ferramentas de *data warehouse*. Por fim, apresentamos algumas especificidades das aplicações OLAP.



REFERÊNCIAS

ELSMARI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Education do Brasil, 2011.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de bancos de dados**. 3. ed. Porto Alegre: McGraw Hill, 2008.