Parte I

Capítulo

2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fundamentação Teórica

2.1 **BUSINESS INTELLIGENCE**

Segundo Rob (2011), o termo *Business intelligence* (**BI**) é utilizado para descrever um conjunto amplo, coeso e integrado de ferramentas e processos utilizados para captar, coletar, integrar, armazenar e analisar dados para a geração e a apresentação de informações que deem suporte à tomada de decisões de negócio. Como o próprio nome diz, **BI** trata da criação de inteligência sobre o negócio. Portanto, o **BI**, é um modelo que permite à empresa transformar dado em informação, informação em conhecimento e conhecimento em sabedoria.

O **BI** não é, por si só, um produto, mas um modelo de conceitos, práticas, ferramentas e tecnologias (*data warehouse*, *data mart*, **OLAP** e/ou ferramentas de mineração de dados) que auxiliam uma empresa a compreender melhor seus recursos centrais e identificam oportunidades fundamentais para criar competitividade (Rob, 2011). Em geral, o **BI** envolve as seguintes etapas:

* Coleta e armazenamento de dados operacionais.
* Agregação de dados operacionais em dados de suporte a decisões.
* Análise de dados de suporte a decisões para gerar informações.
* Apresentação dessas informações ao usuário final para dar suporte a decisões de negócios.
* Tomada de decisões de negócio, o que, por sua vez, gera mais dados que são coletados, armazenados etc. (reiniciando o processo).
* Monitoramento para avaliar os resultados das decisões de negócio (Rob, 2011).

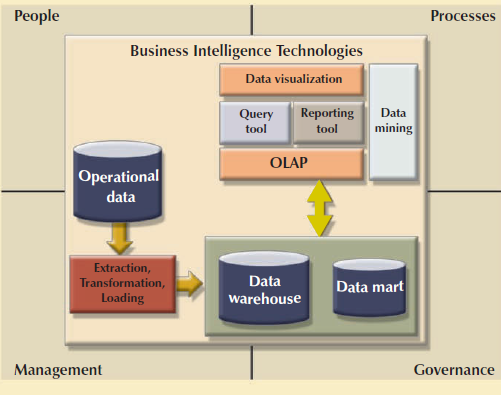
Para implementar todas essa etapas, o **BI** utiliza diversos componentes e tecnologias, o que será apresentado nas seções a seguir.

**2.1.1 Arquitetura de Business Intelligence**

Segundo Rob (2011), o **BI** utiliza-se de tecnologias e aplicações para o gerenciamento de todo o ciclo de vida dos dados, da aquisição ao armazenamento, transformação, integração, análise, monitoramento e apresentação. Não existe uma arquitetura única de **BI**, no entanto, há alguns tipos gerais de recursos, que são compartilhados por todas as implementações de **BI**.

Como poderia deixar de ser, uma boa infraestrutura de *TI*, sempre é composta de dados, pessoas, processos, tecnologias e gerenciamento desses componentes e, para uma arquitetura de **BI**, não pode ser diferente. A Figura 2.1 ilustra um modelo de BI.

**Figura 2.1**. Modelo de *business intelligence*. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



Para compreender a arquitetura de **BI**, será feita uma descrição dos

componentes básicos que fazem parte de sua infraestrutura. Alguns desses componentes, possuem recursos adicionais. Porém, há quatro componentes básicos que todos os ambientes de **BI** devem fornecer, descritos as seguir (Rob, 2011):

* **Ferramentas de extração, transformação e carregamento (ETL) de dados**: esse componente é encarregado de coletar, filtrar, integrar e agregar dados operacionais a serem salvos em um armazém de dados otimizado para o suporte a decisões.
* **Armazenamento de dados**: o armazém de dados é otimizado para o suporte a decisões e costuma ser representado por um *data warehouse* ou data mart. Ele contém dados de negócios extraídos de bancos de dados operacionais e de fontes externas. Esses dados são armazenados em estruturas otimizadas, com foco na velocidade de análise e consulta.
* **Ferramentas de consulta e análise de dados**: esse componente executa as tarefas de recuperação, análise e mineração, utilizando os dados no armazém de dados e os modelos de análise de dados de negócio. Tal componente é utilizado pelo analista de dados para criar as consultas que acessam o banco de dados. Essa ferramenta orienta o usuário sobre quais dados selecionar e como construir um modelo de dados confiáveis. Tal componente costuma aparecer na forma de uma ferramenta **OLAP**.
* **Ferramentas de apresentação e visualização de dados**: esse componente é encarregado de apresentar os dados ao usuário final de várias formas. É utilizado pelo analista de dados para organizar e apesentar os dados. Essa ferramenta ajuda o usuário final a selecionar o formato de apresentação mais adequado, como relatório resumido, mapa ou gráfico.

Deve-se ficar atento para o fato de que, os bancos de dados de suporte a decisões tendem a ser muito grandes. Muitos chegam à faixa dos gigabytes ou terabytes. Esses bancos de dados, demandam por análise sofisticada de dados e, incentivaram a criação de um novo tipo de armazém de dados. Esse armazém de dados, contém dados em formatos que facilitam sua extração, análise e a tomada de decisões. É conhecido como *data warehouse* e se tornou o fundamento de uma nova geração de sistemas de tomada de decisões (Rob, 2011).

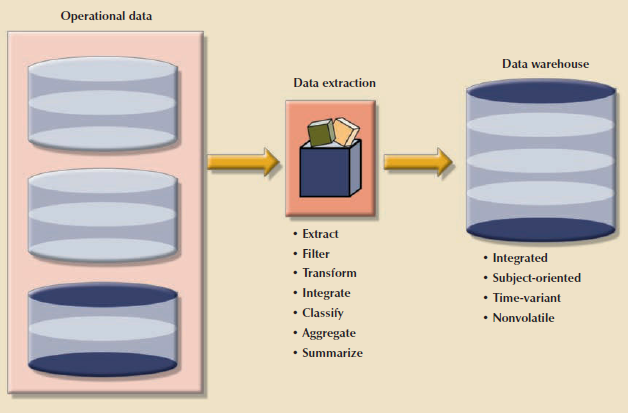
**2.2 DATA WAREHOUSE**

Segundo Inmon (1994), o termo ***data warehouse*** é “um conjunto de dados integrado, orientado por assunto, variável no tempo e não volátil, que fornece suporte a tomada de decisões”. A seguir, será detalhado cada um desse componentes (Inmon, 1994):

* **Integrado**. O *data warehouse* é um banco de dados consolidado e centralizado, que integra dados proveniente de toda a organização e de várias fontes, com diversos formatos.
* **Orientado por assunto**. Os dados do *data warehouse* são dispostos e otimizados de modo a fornecerem respostas a perguntas provenientes de diversas áreas funcionais da empresa. São organizados e resumidos por temas, contendo assuntos de interesse especifico – produtos, clientes, departamentos, regiões, promoções, e assim por diante.
* **Variável no tempo**. Os sistemas transacionais focam nas transações correntes, enquanto os sistemas de *data warehouse*, representam o fluxo de dados através do tempo. Ou seja, os dados são carregados periodicamente no *data warehouse*, e quando isso acontece, todas as agregações dependentes do tempo (ouse dependentes dessa carga de dados), são recalculadas. Por exemplo, se os dados de vendas da semana, são carregados no *data warehouse*, serão atualizadas todas as agregações dependentes dessa carga, ou seja, os agregados semanais, mensais, anuais e de qualquer outras periodicidade que seja dependente dessa carga. Cada conjunto de dados, ao ser carregado em um *data warehouse*, fica vinculado a um rótulo temporal que o identifica dentre os demais. Cada rótulo temporal, fica portanto, associado a uma visão instantânea e sumarizada dos dados operacionais que corresponde ao momento de carga do *data warehouse*. Dessa forma, na medida que o data warehouse vai sendo carregado com tais visões, pode-se realizar análise de tendências a partir dos dados.
* **Não volátil**. Uma vez inserido um dado no *data warehouse*, ele nunca será removido. Uma vez que ele representa o histórico da empresa. Por este fato, o *data warehouse* está sempre crescendo. Portanto, o SGBD, que dá suporte a ele, deve ser capaz de suportar vários gigabytes de dados, ou até mesmo tera-bytes, operando com hardware com diversos processadores.

Resumindo, o *data warehouse* é um repositório de dados semanticamente consistente, que serve como uma implementação física de um modelo de dados de apoio a decisões. Ele armazena as informações que uma empresa necessita para tomar decisões (Han & Kamber, 2011). Normalmente é um banco de dados apenas de leitura, otimizado para processamento de análises e consultas. Em geral, os dados são extraídos de diversas fontes e, em seguida, transformados e integrados, antes de serem carregados no *data warehouse* (Inmon, 1994). A Figura 2.2 ilustra como o data warehouse é criado a partir dos dados contidos em um banco operacional.

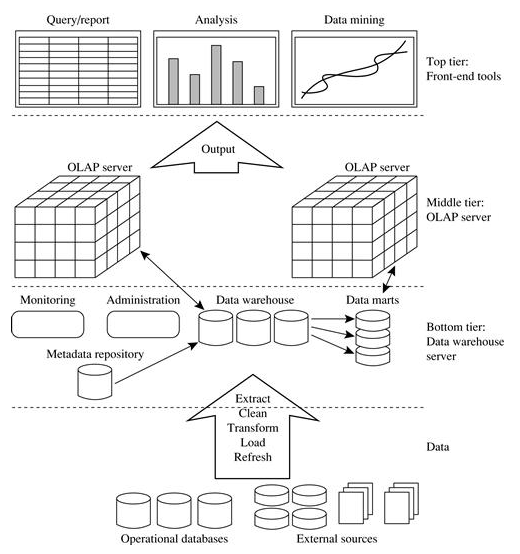
**Figura 2.2**. Criação de um **data warehouse**. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**2.2.1 Arquitetura de Data Warehouse**

Segundo Han & Kamber (2011), um **data warehouse** adota uma arquitetura em três camadas, como ilustra a Figura 2.3.

**Figura 2.3**. Uma arquitetura de data warehouse em três camadas. Fonte: Han & Kamber. Data Mining: Concepts and Techniques. 3nd, 2011.



1. A camada inferior (*bottom*) é um servidor de *data warehouse*, que quase sempre, é um sistema de banco de dados relacional. Segundo Han & Kamber (2011), são usadas ferramentas de back-end e utilitários para extrair dados dessa camada e alimentar a camada superior. Ainda segundo Han & Kamber (2011), os dados são extraídos usando interface de programação de aplicativo, conhecidos como ***gateways***. Um *gateway* permite que clientes gerem código SQL, para ser executado no servidor. Pode-se citar como exemplos de *gateways* ODBC (*Open Database Connection*) e OLEDB (*Object Linking and Embedding Database*) da Microsoft e JDBC (*Java Database Connection*). Essa camada também contém um repositório metadata, o qual armazena informações sobre o *data warehouse* e seus conteúdos.
2. A camada intermediária (*middle tier*), segundo Han & Kamber (2011), é um servidor OLAP que geralmente é implementado usando (1) um modelo relacional **OLAP** (**ROLAP**) (fornece recursos de OLAP utilizando bancos de dados relacionais e ferramentas familiares de consulta relacional para armazenar dados multidimensionais; ou (2) um modelo multidimensional **OLAP** (**MOLAP**) (amplia os recursos de OLAP para sistemas de gerenciamento de banco de dados multidimensionais (**SGBDM**s). O SGBDM utiliza técnicas especiais para armazenar dados em matrizes de n dimensões. O pressuposto do MOLAP é que os bancos de dados multidimensionais são os mais adequados para gerenciar, armazenar e analisar dados multidimensionais (Rob, 2011).
3. A terceira camada (top), segundo Han & Kamber (2011), é o fronte-end do cliente, a qual contém as ferramentas de consulta, de relatório, de análise e mineração de dados (por exemplo, análise de tendência, previsão, e assim por diante.).

Segundo Han & Kamber, do ponto de vista da arquitetura, há três modelos de data warehouse: o warehouse empresarial, o data mart e warehouse virtual.

**2.2.2 Data Warehouse Empresarial**

Um data warehouse empresarial, segundo Han & Kamber (2011), coleta todas as informações sobre todos os assuntos da organização. Ele fornece informações de dados de toda empresa, geralmente, oriundas de um ou mais sistemas operacionais ou informações obtidas externamente. Ele contém dados detalhados e/ou sumarizados, e pode variar em tamanho de poucos gigabytes para centenas de gigabytes, terabytes ou superiores. A sua implementação pode ser em um mainframe, supercomputador, em uma plataforma de arquitetura paralela. Requer uma modelagem comercial extensiva e pode levar muitos anos para ser projetado e construído (Han & Kamber, 2011).

**2.2.3 Data Mart**

De acordo com Inmon (1994), embora o *data warehouse*, seja uma proposta muito atraente, que traga muitos benefícios, os gerentes podem relutar em adotar essa estratégia, pelo fato de que, a criação de um *data warehouse* exige tempo, dinheiro e considerável esforço gerencial. Estes fatos, fazem com que muitas empresas iniciem na criação de *data warehouse*, focando em conjuntos de dados gerenciais, orientados a atender pequenas áreas de negócio, dentro da empresa. Esses armazenamentos menores são chamados de ***data marts***. Um ***data mart*** é portanto, segundo Inmon (1994), um pequeno subconjunto de um *data warehouse*, sobre um único assunto, que fornece suporte às decisões de um pequeno grupo de pessoas. No entanto, pode-se criar um *data mart* a partir de dados extraídos de um *data warehouse*, com a finalidade especifica de dar suporte a um acesso mais rápido a determinado grupo ou função. Dessa forma, os *data marts* e o *data warehouse* podem coexistir em um ambiente de business *intelligence* (Inmon, 1994).

**2.2.4 Virtual Data Warehouse**

De acordo com Han & Kamber (2011), um warehouse virtual é um conjunto de visões sobre bases de dados operacionais. Você pode materializar algumas visões operacionais, para obter um processamento de consultas eficientes. O warehouse virtual é o estado de visibilidade global de recursos, com base na aquisição e processamento de dados operacionais em tempo real. Informações disponíveis no armazém virtual tem o potencial de reduzir custos e melhorar o serviço ao cliente. A infraestrutura já está disponível para captura de dados em tempo real, e o custo de aquisição de dados continuará a reduzir.

**2.3 PROCESSAMENTO ANÁLITICO ON-LINE**

De acordo com Rob (2011), a necessidade de suporte a decisões mais intensivo, levou à introdução de uma nova geração de ferramentas. Tais ferramentas, foram denominadas de **processamento analítico on-line** (**OLAP** – *Online Analytical Processing*). Essa nova ferramenta cria um ambiente avançado de análise de dados que dá suporte à tomada de decisões, modelagem comercial e pesquisa operacional. Ainda segundo Rob (2011), esses sistemas comportam quatro características principais:

* Utilizam técnicas de análise de dados multidimensionais.
* Proporcionam suporte avançado a bancos de dados.
* Fornecem interface fácil de utilizar para o usuário final.
* Dão suporte a arquitetura cliente/servidor.

**2.3.1 Técnicas de análise de dados multidimensionais**

De acordo com Rob (2011), a característica mais evidente das modernas ferramentas **OLAP**, é a capacidade de análise multidimensional, onde, os dados são processados e visualizados como parte de uma estrutura multidimensional. Essas técnicas de análise de dados multidimensionais, utilizam as seguintes funções:

* Funções avançadas de apresentação de dados. Gráficos 3D, pivô, tabulações cruzadas, rotação de dados e cubos tridimensionais.
* Funções avançadas de agregação, consolidação e classificação de dados. Permitem que o analista de dados crie vários níveis de agregação, detalhamento de dados, *drill down* e *roll up* de dados em diferentes dimensões e níveis.
* Funções computacionais avançadas. Incluem variáveis orientadas para negócio (tais como participação de mercado, margem de vendas, etc.), relações financeiras e contábeis (tais como lucratividade, despesas gerais, alocação de custos e retorno), funções estatísticas e de previsão.
* Funções avançadas de modelagem de dados. Dão suporte para cenários de simulação, avaliação de variáveis, contribuições de variáveis para o resultado, programação linear, dentre outras, ferramentas de modelagem.

**2.3.2 Suporte Avançado de Banco de Dados**

As ferramentas **OLAP**, para apresentar suporte eficiente a decisões, deve ter recursos avançados de acesso a dados. Tais recursos incluem (Rob, 2011):

* Acesso a vários tipos de SGBDs, arquivos fora do banco de dados (flat-file) e fontes de dados internas e externas.
* Acesso a dados agregados de *data warehouse*.
* Recursos avançados de navegação de dados, como *drill down* e *roll up*.
* Tempo rápido e consistente de resposta a consultas.
* Capacidade de mapear solicitações de usuários finais, para a fonte adequada de dados e, em seguida, para a linguagem adequada de acesso aos dados (normalmente SQL). Segundo Rob (2011), as ferramentas OLAP mapeiam os elementos de dados do *data warehouse* e do banco de dados operacional em seus próprios dicionários. Esses metadados são utilizados para traduzir as solicitações de análise de dados do usuário final em código de consulta (otimizados) adequados que são, em seguida, direcionados para a(s) fonte(s) de dados correta(s).
* Suporte a bancos de dados muito grandes. Pois como já foi falado, o *data warehouse* pode crescer rapidamente e atingir facilmente gigabytes ou até mesmo terabytes.

**2.3.3 Interface fácil de utilizar para os usuários finais**

De acordo com Rob (2011), os recursos avançados OLAP são mais úteis quando o acesso a eles são simples. Então, os fornecedores conhecedores dessas necessidades, supriram suas ferramentas OLAP, com interfaces gráficas fáceis de utilizar. Muitos desses recursos foram herdados de ferramentas de análise de dados anteriores, já familiares aos usuários finais.

**2.3.4 Arquitetura Cliente/Servidor**

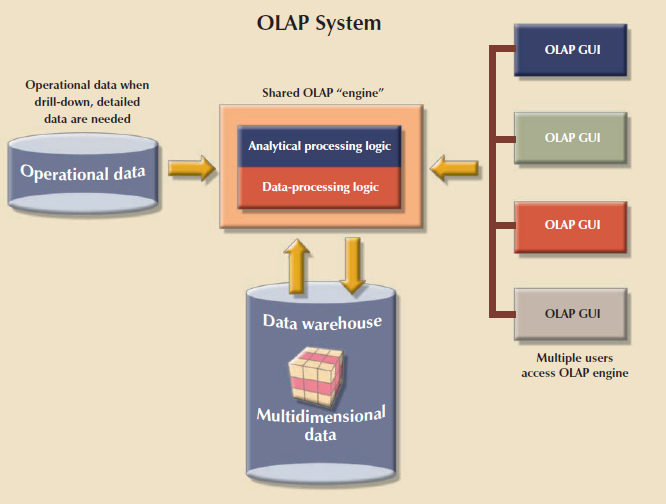
Um ambiente cliente/servidor possibilita que um sistema OLAP seja dividido em vários componentes que definem sua arquitetura. Esses componentes podem, então, ser colocados no mesmo computador ou distribuídos entre diversas máquinas. Assim, segundo Rob (2011), o OLAP é projetado para atender a exigências de facilidades de utilização, ao mesmo tempo em que mantém a flexibilidade do sistema.

**2.3.5 Arquitetura OLAP**

Segundo Rob (2011), os sistemas OLAP são projetados para utilizar tanto dados operacionais como de **data warehouse**. Para isso, existem várias arquiteturas de instalação de sistema OLAP, baseados nas regras de negócio de cada empresa. Por exemplo, existe situação onde o sistema OLAP é instalado em um único computador, o problema nessa arquitetura, é que, o computador deve ser potente para armazenar o sistema OLAP e executar localmente todos os processamentos de dados. Uma outra arquitetura é, aquela em que o GUI do OLAP executa em estações de trabalho clientes, enquanto o mecanismo OLAP, ou servidor, composto da lógica de processamento analítico e de processamento de dados, é executado em computador compartilhado. Nesse caso, o servidor será um *front end* para todos os dados de suporte a decisões do *data warehouse*.

O fato é, segundo Rob (2011), que um sistema OLAP pode acessar ambos tipos de armazenamento de dados (operacional ou *data warehouse*) ou apenas um, dependendo da implementação que se deseje configurar. Em todo caso, a análise multidimensional de dados exige algum tipo de representação de dados multidimensionais, o que normalmente é fornecido pelo mecanismo OLAP. A Figura 2.4 ilustra uma arquitetura OLAP com organização de armazenagem de dados multidimensionais.

**Figura 2.4** Servidor OLAP com organização de armazenagem de dados multidimensionais. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



De acordo com Rob (2011), há diversas formas de gerenciar e armazenar os dados em um sistema OLAP, como já foi visto anteriormente, o OLAP relacional (**ROLAP**) e o OLAP multidimensional (**MOLAP**).

**2.3.5.1** **O processamento analítico on-line relacional** (**ROLAP**, sigla em inglês para *Relational Online Analytical Processing*).

Segundo Rob (2011), essa abordagem se estrutura a partir de tecnologias relacionais existentes e representa uma extensão natural para todas as empresas que já utilizem sistemas de gerenciamento de banco de dados relacionais. O ROLAP utiliza uma técnica especial de projeto que permite à tecnologia SGBDR dar suporte a representações de dados multidimensionais, conhecida como “Esquema estrela”. O esquema estrela é essencial para o sistema ROLAP, uma vez que, a tecnologia relacional utiliza a normalização como metodologia de projetos de banco de dados e, isso é visto como um obstáculo em sistemas OLAP. Como se sabe, a normalização divide as entidades de negócio em partes menores para produzir tabelas normalizadas. O motivo de se usar tabelas normalizadas é reduzir as redundâncias, eliminando as anomalias. No entanto, para sistemas de suporte a decisões, é mais fácil compreender os dados, quando os mesmos, são vistos em relação a outros dados.

Um outro fator dos sistemas relacionais é que a SQL não é adequada para executar análise avançada de dados. Segundo Rob (2011), a maioria das solicitações de dados de suporte a decisões, exige o uso de consultas de SQL *multiple pass* (capaz de fazer várias passagens de processamento) ou vários comandos de SQL integrada. Portanto, segundo Rob (2011), para resolver este problema, as tecnologias ROLAP estende a SQL de modo que ela possa diferenciar entre exigências de acesso a dados de data warehouse (baseado em esquema estrela) e dados operacionais (tabelas normalizadas). Desse modo, o sistema ROLAP é capaz de gerar o código SQL necessário para acessar dados de esquema estrela.

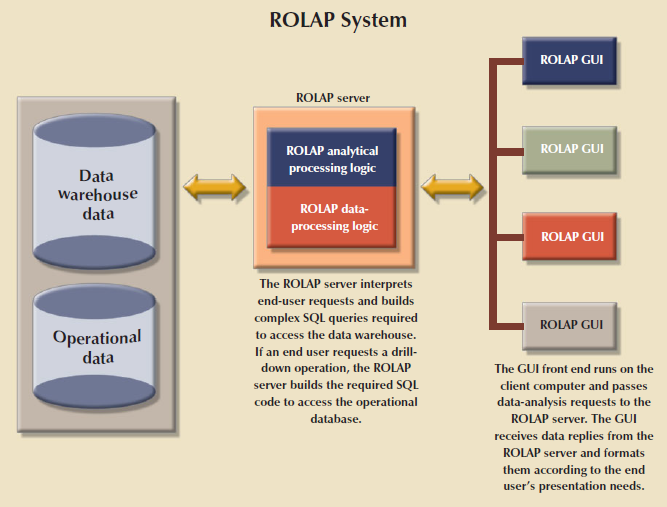
As ferramentas de ROLAP são produtos Cliente/Servidor em que a interface do usuário final, o processamento analítico e o processamento de dados ocorrem em computadores diferentes. A Figura 2.5 ilustra a interação dos componentes de ROLAP cliente/servidor.

2.3.5.2 **O processamento analítico on-line multidimensional** (**MOLAP**, sigla em inglês para Multidimensional *Online Analytical Processing*).

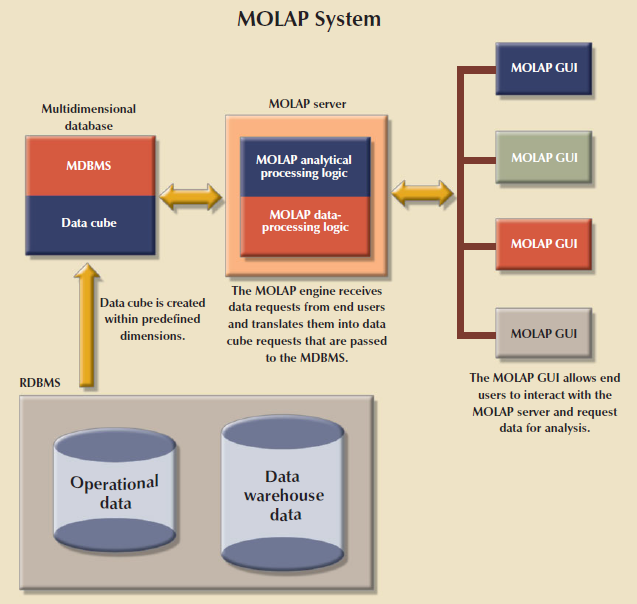
De acordo com Rob (2011), o MOLAP amplia os recursos de OLAP para sistemas de gerenciamento de banco de dados multidimensionais (**SGBDMs**). O pressuposto do MOLAP é que os bancos de dados multidimensionais são os mais adequados para gerenciar, armazenar e analisar dados multidimensionais.

Conceitualmente, os usuários finais de SGBDM visualizam os dados armazenados como um **cubo de dados**. Em um cubo de dados, a localização de cada valor de dado, é uma função dos eixos x, y e z em um espaço tridimensional. Os eixos x, y e z representam as dimensões do valor do dados. Os cubos podem crescer até um número de n dimensões, tornando-se, assim, *hipercubos*. De acordo com Rob (2011), uma característica importante dos cubos é que são estáticos, ou seja, não estão sujeitos a alterações e devem ser criados antes de sua utilização. Um outro fato é, que eles não podem ser criados por consultas ad hoc. Em vez disso, a consulta é feita em cubos pré-criados com eixos definidos. Por exemplo, um cubo de vendas terá as dimensões de produto, localização e tempo, permitindo a consulta apenas dessas dimensões. A Figura 2.6 ilustra uma arquitetura cliente/servidor de MOLAP.

**Figura 2.5** Arquitetura ROLAP de cliente/servidor. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**Figura 2.6** Arquitetura MOLAP de cliente/servidor. **Fonte** Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**Tabela 2.1** Resume algumas das principais diferenças entre OLAP e MOLAP.

|  |  |  |
| --- | --- | --- |
| CARACTERISTICA | ROLAP | MOLAP |
| Esquema | Utiliza esquema estrela  É possível acrescentar novas dimensões dinamicamente. | Utiliza cubos de dados  Dimensões adicionais exigem a recriação do cubo de dados. |
| Tamanho do banco de dados | Médio a grande | Pequeno a médio |
| Arquitetura | Cliente/Servidor com base em padrões aberto | Cliente/Servidor  Proprietário |
| Acesso | Suporte a solicitações ad hoc  Dimensões ilimitadas | Limitado a dimensões predefinidas |
| Recursos | Altos | Muito altos |
| Flexibilidade | Alta | Baixa |
| Escalabilidade | Alta | Baixa |
| Velocidade | Boa com pequenos Data Sets;  Razoável para data sets médios ou grandes | Mais rápido com data sets pequenos ou médios; |

**2.4 MODELAGEM MULTIDIMENSIONAL**

A modelagem multidimensional é uma forma de Modelagem de Dados voltada para concepção e visualização de conjunto de medidas que descrevem aspectos comuns de um determinado assunto. É utilizada especialmente para sumarizar e reestruturar dados, apresentando-os em visões que suportem a análise dos dados envolvidos (Passos & Goldschimdt, 2005).

De acordo com Han & Kamber (2011), o *data warehouse* e as ferramentas **OLAP** são baseadas em um **modelo de dados multidimensional**. Nesse modelos, os dados são visto na forma de um cubo de dados. Um modelo multidimensional possui três componentes básicos: Fatos (***facts tables***), Dimensões (***dimensions***) e Medidas (***measures***). E existem diversas formas de modelagem física de um data warehouse, incluindo esquema estrela (***star schema***), esquema floco de neves (***snowflake***) e constelação de fatos (***fact constellation***). A seguir serão discutidas cada uma desses conceitos.

**2.4.1 Esquema Estrela**

O esquema estrela, segundo Rob (2011), é uma técnica de modelagem de dados multidimensionais de suporte a decisões em um banco de dados relacional. Ainda segundo, Rob (2011), o esquema estrela foi desenvolvido, pois as técnicas de modelagem relacional, entidade relacionamento (**ER**) e normalização existentes não produziam uma estrutura que atendesse às necessidades de análise avançada de dados.

O modelo estrela é fácil de implementar e, ao mesmo tempo em que preserva as estruturas relacionais, em que o banco operacional criado. O esquema estrela básico possui quatro componentes: fatos, dimensões, atributos e hierarquias de atributos.

**2.4.1.1 Fatos**

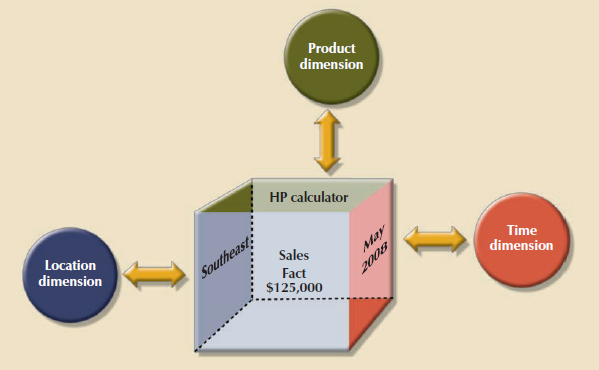
Um **fato** é uma coleção de itens de dados, composta de dados de medidas e de contexto. Representa um item, ou uma transação ou um evento associado ao tema da modelagem. São medidas numéricas (valores) que representam um aspecto ou atividade específica dos negócios. Os fatos normalmente utilizados em análise de dados comerciais são unidades, custos, preços e receitas. Os fatos são armazenados em tabelas de fatos que constituem o centro do esquema estrela. A **tabela de fatos** (***fact table***) contém fatos vinculados por meio de suas dimensões (Kimball, 2002, Passos & Goldschimdt, 2005).

Segundo Rob (2011), os fatos também podem ser computados ou derivados no momento da execução. Esses as vezes são chamados de métricas para diferenciá-los dos fatos armazenados.

**2.4.1.2 Dimensões**

Uma dimensão é um tipo de informação que participa da definição de um fato. As dimensões determinam o contexto do assunto. As **dimensões** são características de qualificação que fornecem perspectivas adicionais a um determinado fato. Os dados de suporte a decisões são quase sempre vistos relacionados a outros dados, por isso, as dimensões são interessantes. Por exemplo, pode-se, em um sistema de suporte a decisões, querer comprar as vendas de certos produtos, entre regiões e entre períodos. Nesse exemplo, teríamos as vendas, as dimensões produto, localização e tempo. Segundo Rob (2011), as dimensões ampliam a visão dos fatos. Essa dimensões são armazenadas em **tabelas de dimensões**. A Figura 2.7 ilustra um esquema estrela para vendas com as dimensões de produto, localização e tempo.

**Figura 2.7** **Esquema estrela**. **Fonte:** Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**2.4.1.3 Medidas**

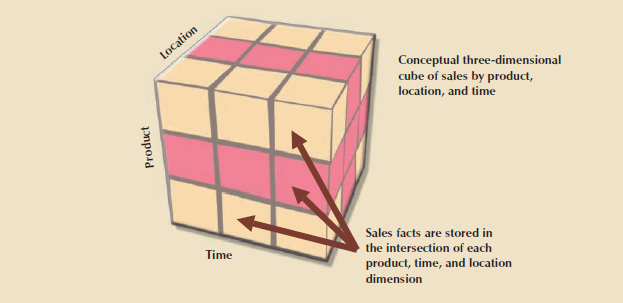
Uma medida é um atributo ou variável numérica que representa um fato. Exemplos: valor da ação, número de evasões escolares, quantidade de produtos vendidos, valor total de venda, etc.

**2.4.1.4 Atributos**

De acordo com Kimball (2002), cada tabela de dimensão contém atributos. Os atributos costumam ser utilizados para buscar, filtrar e classificar fatos. As dimensões fornecem características descritivas sobre os fatos por meio de seus atributos.

Segundo Rob (2011), conceitualmente, o modelo de dados multidimensional do exemplo de vendas, é melhor representado por um cubo tridimensional. Isso, não significa que haja um limite para o número de dimensões, que podem ser associadas a uma tabela de fatos. Não há limite matemático para o número de dimensões utilizadas. Usar um modelo tridimensional, torna mais fácil a visualização do problema. A Figura 2.8 ilustra uma visão das vendas dimensionadas por produto, localização e tempo, na terminologia de análise de dados multidimensional.

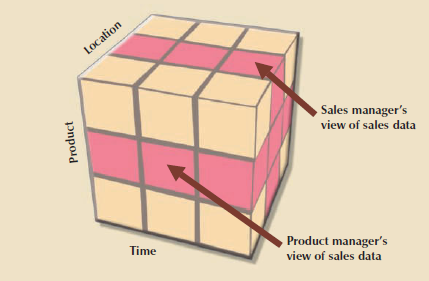
**Figura 2.8 Visão tridimensional de vendas**. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



Observe que cada valor de venda armazenado no cubo da Figura 2.8 está associado às dimensões de localização, produto e tempo. Lembre-se que isso é apenas uma representação conceitual de dados multidimensional e não mostra como os dados estão fisicamente armazenados no *data warehouse*. Os mecanismos **ROLAP** e **MOLAP** têm suas próprias tecnologias para simular esse cubo multidimensional (Rob, 2011).

Um dos principais recursos da análise multidimensional, segundo Rob (2011), é a sua capacidade de focar em “fatias” específicas dos cubo. Em termos, multidimensionais, essa capacidade do cubo em executar uma análise mais detalhada é conhecida com detalhamento. A Figura 2.9 ilustra esse conceito. Observando-se a Figura 2.9, ver-se que cada corte através do cubo produz uma fatia. A interseção de fatias produz pequenos cubos de dados.

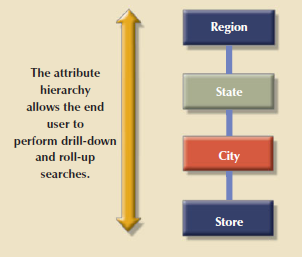
**Figura 2.9** **Visão de detalhamento de vendas**. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**2.4.1.5 Hierarquias de Atributos**

De acordo com Kimball (2002), os atributos no interior de dimensões podem ser ordenados em hierarquias bem definidas. A hierarquia de atributos fornecem uma organização vertical utilizada para duas finalidades principais: agregação e análise de dados por ***drill down*** e ***roll up***. A Figura 2.10 ilustra como os atributos da dimensão de localização podem ser organizados em uma hierarquia por região, estado, cidade e loja.

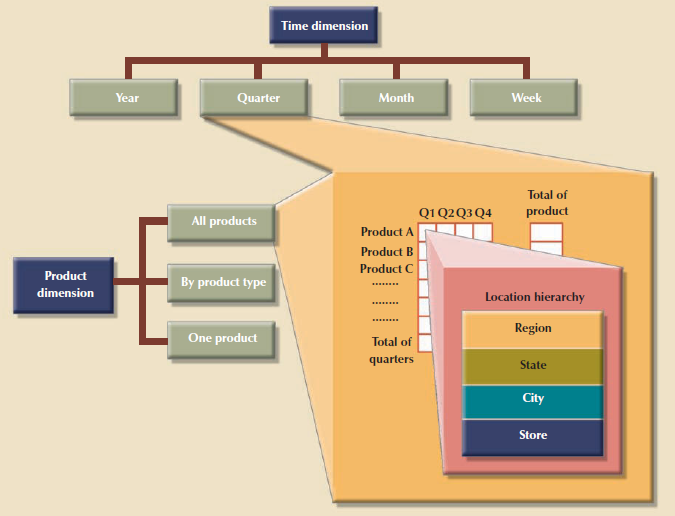
**Figura 2.10** Hierarquia de atributo de localização. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



De acordo com Rob (2011), a hierarquia de atributos fornece a possibilidade de executar buscas de *drill down* e *roll up* no *data warehouse*. A Figura 2.11 ilustra um cenário em que o analista de dados estuda fatos de vendas utilizando as dimensões de produto, tempo e localização. Nesse exemplo, a dimensão de produto é estabelecida como “Todos os produtos”, o que significa que o analista de dados verá todos os produtos no eixo y. a dimensão de tempo (eixo x) é estabelecida com “Trimestre”, indicando que os dados estão agregados por trimestre (por exemplo, vendas totais dos produtos A, B e C em T1, T2, T3 e T4). Finalmente, a dimensão de localização é inicialmente estabelecida para “Região”, garantindo, assim, que cada célula contenha as vendas regionais totais de determinado produto em determinado trimestre.

A tabela de fatos é relacionada com cada tabela de dimensões em um relacionamento de “muitos para um” (M:1). Dito de outra forma, várias linhas de fatos se relacionam a cada linha de dimensão. Deve-se ser lembrado que, os fatos e as dimensões são normalmente representadas por tabelas físicas no banco de dados do *data warehouse*. A Figura 2.12 ilustra os relacionamentos entre as tabelas de fatos de vendas e as tabelas das dimensões de produto, localização e tempo. Além dos mais, foi adicionada uma dimensão de cliente ao modelo, para mostrar como o esquema estrela pode ser expandido facilmente.

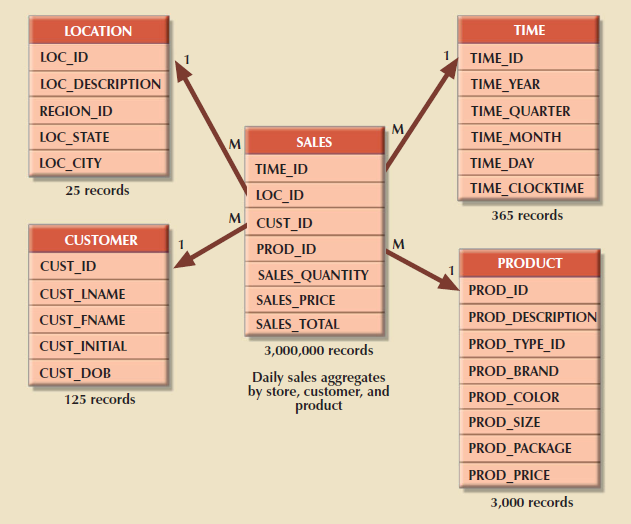
**Figura 2.11** Hierarquias de atributos em análise multidimensional. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



De acordo com Kimball (2002), a criação de um banco de dados que forneça respostas rápidas e precisas a consultas de análise de dados, é o principal objetivo do *data warehouse*. Dessa forma, existem medidas de aprimoramento de desempenho que objetivam a velocidade das consultas, facilitando o código SQL e melhorando a representação semântica das dimensões de negócio. Segundo Kimball (2002), costuma-se utilizar quatro técnicas para otimizar o projeto de *data warehouse*:

* Normalização de tabelas dimensionais.
* Manutenção de várias tabelas de fatos para representar diferentes níveis de agregação.
* Desnormalização de tabelas de fatos.
* Particionamento e replicação de tabelas.

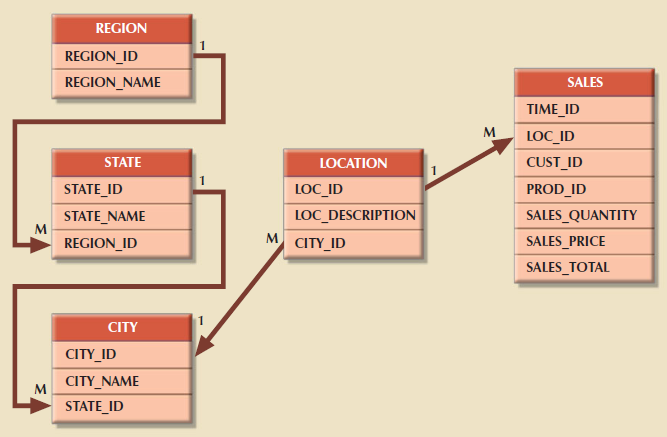
**Figura 2.**12 Esquema estrela para Vendas. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**2.4.2 Esquema Floco de Neves**

Segundo Rob (2011), para facilitar a navegação do usuário final, utiliza-se a técnica de normalização das tabelas dimensionais. Esse esquema normalizado é conhecido como esquema floco de neves, que nada mais é do que, um tipo de esquema estrela, no qual as tabelas de dimensões podem ter suas próprias tabelas de dimensões. Resumindo, o esquema floco de neve resulta normalmente da normalização de tabelas de dimensão. Por exemplo, se a tabela da dimensão de localização contém dependência transitivas entre região, estado e cidade, é possível normalizar esta tabela dimensão para a 3FN (terceira forma normal). A Figura 2.13 ilustra essa normalização.

**Figura 2.13** **T**abelas de dimensões normalizadas. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



Essa normalização, simplifica as operações de filtragem de dados relacionados a dimensão. No entanto, há um preço a pagar por ela, pois aumenta-se a complexidade das consultas SQL. Por exemplo, caso se deseje agregar os dados por região, deve-se utilizar uma junção de quatro tabelas.

De acordo com KIMBALL (2002), também é possível acelerar as operações de consultas criando e mantendo várias tabelas de fatos relacionadas a cada nível de agregação. KIMBALL (2002), define isso como uma **constelação de fatos**. Por exemplo, uma solução usando essa técnica, para o problema apresentado na Figura 2.13, seria criar uma tabela de fato para Vendas\_Região, Vendas\_Estado, Vendas\_Cidade e Vendas\_Localização. Essas tabelas agregadas são computadas previamente na fase de carregamento de dados, e não no momento da execução (Rob, 2011). A finalidade dessa técnica, segundo Rob (2011), é poupar ciclo do processador durante a execução, acelerando, assim, a análise de dados.

De acordo com Passos e Goldschmidt (2005), existem diversos operadores OLAP que permitem acessar os dados em modelos multidimensionais. A seguir encontra-se indicados alguns deles:

* **Drill up/down** – Utilizado para amentar ou reduzir o nível de detalhe da informação acessada. Exemplo: Vendas por país, Vendas por estado, etc.
* **Slicing** – Utilizado para selecionar as dimensões a serem consideradas na consulta. Exemplo: Visualizar as vendas, separadas por país e por mês.
* **Dicing** – Utilizado para limitar o conjunto de valores a ser mostrado, fixando-se algumas dimensões. Exemplo: Vendas de um determinado estado, de um determinado produto em um determinado ano.
* **Pivoting** – Utilizado para inverter as dimensões entre linhas e colunas. Exemplo: Ao visualizar vendas por produto e por estado, aplicar o operador para visualizar as vendas por estado e por produto.
* **Data Surfing** – Executar uma mesma análise em outro conjunto de dados. Exemplo: Ao avaliar as vendas no Brasil, aplicar o operador para realizar a mesma consulta em Portugal.

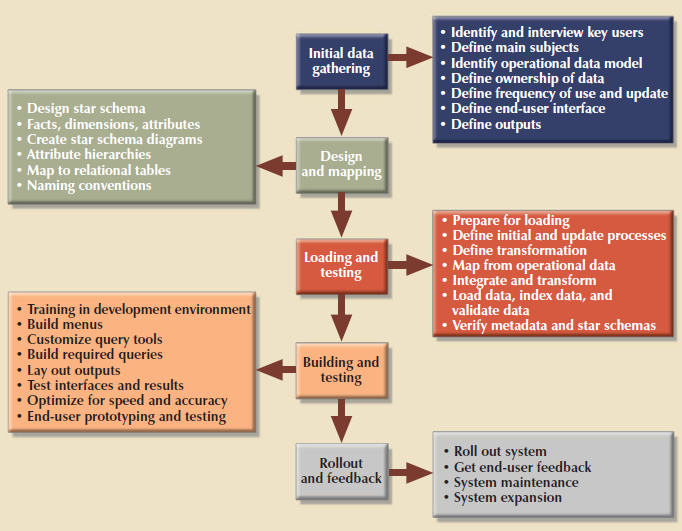
Ainda de acordo com Rob (2011), quando um sistema de BI é implementado em áreas geograficamente dispersas, as técnicas de particionamento e replicação são especialmente importantes. O **particionamento** separa a tabela em subconjunto de linhas ou colunas e coloca esses subconjuntos próximos ao computador cliente, melhorando, dessa forma, o tempo de acesso, por outro lado, a **replicação** faz uma cópia da tabela e a coloca em uma localização diferente, também com a finalidade de aprimorar o tempo de acesso.

Resumindo, projetar um *data warehouse* significa receber a oportunidade de ajudar a desenvolver um modelo integrado que capture os dados considerados essenciais para a organização, tanto da perspectiva do usuário final, como da perspectiva dos negócios. Para tanto, um projeto de *data warehouse*, deve satisfazer:

* Critérios de integração e carregamento de dados.
* Recursos de análises de dados com desempenho aceitável de consulta.
* Necessidades de análises de dados do usuário final

Segundo Rob (2011), a preocupação técnica mais evidente na implementação de um *data warehouse* é fornecer ao usuário final suporte a decisões com recursos avançados de análise de dados – no momento certo, no formato certo, com os dados certos e ao custo certo. A Figura 2.14 ilustra um processo simplificado de implementação de *data warehouse*.

**Figura 2.14** Mapa do processo de projeto e implementação de data warehouse. **Fonte**: Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.



**2.5 Conclusões**

Neste capítulo foi apresentada uma visão geral sobre *Business Intellingence* (**BI**), onde deu para perceber que, o BI é um conjunto amplo, coeso e integrado de ferramentas e processos utilizados para captar, coletar, integrar, armazenar e analisar dados para a geração e a apresentação de informações que deem suporte à tomada de decisões para os gestores da empresa.