



DATA ANALYTICS (DW/DM/BI) E BIG DATA

Prof. Msc. Marcos Alexandruk

alexandruk@uni9.pro.br

<https://github.com/alexandruk/dw>



Aula 01

Apresentação da disciplina

Introdução

Data Warehouse se tornou uma realidade em muitas empresas e a cada dia mais empresas adotam sistemas de **Inteligências de Negócios** para tomada de decisões, visando aumentar o lucro.

Com o crescimento da demanda por este tipo de tecnologia, cresce também a necessidade da formação de **novos profissionais** capacitados a suportar e desenvolver este segmento, então vamos estudar.

Introdução

Estamos vivenciando a era do conteúdo, em que aumentar o capital intelectual de uma empresa é uma necessidade competitiva. As organizações que usam, de forma eficiente, a tecnologia da informação adquirem conhecimento e velocidade para alcançar um diferencial nos mercados em que atuam e se tornam capazes de alavancar seus negócios.

Introdução

O ambiente de dados para suportar os processos gerenciais e de tomada de decisão é diferente do ambiente convencional de processamento de transações.

Data Warehouse tem o objetivo de integrar as informações dos principais processos empresariais e possui a capacidade de consolidar tais dados, adquiridos de diferentes acervos, para fins de exploração e análise, ampliando o conteúdo informacional a fim de atender às expectativas e necessidades de nível gerencial e estratégico na empresa.

Introdução

As empresas dependem de sua capacidade de analisar, planejar e reagir, de forma rápida e imediata, às mudanças nas condições de seus negócios. Para que isso aconteça, é necessário que a organização disponha de informações em quantidade e qualidade.

Introdução

Diariamente, dados sobre os mais variados aspectos dos negócios da empresa são gerados e armazenados, e passam a fazer parte dos recursos de informação dessa empresa.

Entretanto, tais informações encontram-se, em geral, espalhadas em diversos sistemas e exigem um esforço considerável de integração para que possam dar suporte efetivo à tomada de decisão de gerentes e executivos.

Por isso, estas decisões normalmente são tomadas com base na experiência dos administradores, quando poderiam também ser baseadas em fatos históricos que foram armazenados pelos diversos sistemas de informação utilizados pelas organizações.

Introdução

Quando desenvolvemos um sistema transacional, seu propósito principal é suportar as transações de negócio da empresa, exemplificando: sistema de vendas, compras, controle de estoque, etc. Estes tipos de sistemas não são projetados para gerar e armazenar as informações estratégicas, por esse motivo, um novo conjunto de conceitos e ferramentas vem ganhando enorme destaque nos últimos anos: a tecnologia de Data Warehouse , que oferece às organizações uma maneira flexível e eficiente de obter as informações necessárias aos seus processos decisórios.

Data Warehouse

As empresas realizam muitas perguntas que não conseguimos responder com sistemas transacionais, como:

- Como reduzir os custos sem impactar na qualidade do produto?
- Quem são os melhores clientes para vender seus produtos?
- Como aumentar a participação de seus produtos no mercado?
- Quais são os segmentos de mercado mais significativos para seu negócio?
- Como maximizar as vendas por cliente?
- Quais promoções realizar?
- Para quais produtos devo realizar promoções?
- Em que momento realizar as promoções?
- Qual o estoque mínimo necessário por produto em determinado período?
- Que produtos têm maior índice de retorno?
- Quais clientes possuem uma conta corrente, mas não uma conta poupança?
- Quais clientes possuem mais de um tipo de conta?
- Quem são os meus clientes mais rentáveis entre os que compram mais?
- Como classificar os clientes por faixas usando métricas analíticas e estatísticas avançadas?

Conhecer a resposta destas e outras perguntas é **vantagem competitiva** que toda empresa deseja, mas normalmente elas desconhecem o potencial dos dados de seu sistema de informação.

Data Warehouse

O Data Warehouse nasceu a partir do reconhecimento da importância do valor da informação nas organizações. Ele é um ambiente expansível e planejado para a análise de dados imutáveis.

Tais dados são lógica e fisicamente transformados, provenientes de múltiplas fontes, atualizados e mantidos por um longo período de tempo, expressos em termos do negócio e resumidos para uma análise eficiente.

Assim, um Data Warehouse é uma plataforma com dados integrados e sua qualidade melhorada para apoiar as tomadas de Decisões pelos executivos das empresas.

Data Warehouse

Em termos tecnológicos, um Data Warehouse é uma combinação de várias tecnologias, tendo como primeiro objetivo a integração efetiva de bases de dados operacionais em um ambiente que habilita o uso estratégico dos dados.

Estas tecnologias incluem sistemas gerenciadores de banco de dados relacional e multidimensional, modelagem de metadados e repositórios, interfaces gráficas para usuário, etc.

Data Warehouse

Algumas definições para Data Warehouse

- É um banco de dados orientado por assunto, integrado, não volátil e histórico, criado para suportar o processo de tomada de decisão, enquanto Data Mart é uma coleção de assuntos, organizados para dar suporte à tomada de decisão e baseados nas necessidades de um determinado departamento.
- É um conjunto de conceitos, métodos e recursos tecnológicos que habilitam a obtenção e distribuição de informações geradas a partir de dados operacionais, históricos e externos, visando proporcionar subsídios para tomadas de decisões gerenciais e estratégicas.
- É uma cópia dos dados da transação; especificamente estruturado para consulta e análise.
- É o processo de integração dos dados corporativos de uma empresa em um único repositório de dados a partir do qual os usuários finais podem facilmente executar consultas, gerar relatórios e fazer análises
- É o resultado de um processo de armazenagem de dados específicos e integrados, de fontes heterogêneas, para a realização de consultas e análises dimensionais.

Data Warehouse

O Data Warehouse surgiu na década de 80 com destaque aos trabalhos de **Bill Inmon**, que tinha o objetivo de resolver a geração de informações empresariais, processo esse que é falho nos bancos de dados transacionais.

Mais conhecido como o "Pai do Data Warehouse", Bill Inmon tornou-se o autor mais conhecido em todo o mundo na área de Data Warehouse/Business Intelligence (mais de 50 livros e 650 artigos). Em 2007, Bill foi nomeado pela Computerworld como uma das "Dez pessoas que importavam de TI nos últimos 40 anos".

Referências

CARVALHO, Luis A. V. **Datamining: A mineração de dados no marketing, medicina, economia, engenharia e administração.** 1. ed. Rio de Janeiro: Ciência Moderna, 2005.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. **Data Mining: um guia prático.** Rio de Janeiro: Campus, 2005.

MACHADO, Felipe Nery Rodrigues. **Tecnologia e projeto de data warehouse.** 2. ed. São Paulo: Érica, 2006.

SILBERSCHATZ, A.; KORTH, H.; SUBARSHAN, S. **Sistema de Banco de Dados.** 5. ed. Rio de Janeiro: Campus, 2006.

TAUB, Benjamin et al; **Oracle 8i data warehouse.** Rio de Janeiro: Campus, 2001.



Aula 02

Introdução e o Ambiente de um Data Warehouse

Objetivo

Entender a evolução do desenvolvimento de um ambiente de banco de dados adequado à análise de negócios e ao apoio à tomada de decisões gerenciais e estratégicas.

Introdução

Data Warehouse é o resultado de um processo de armazenagem de dados específicos e integrados, de fontes heterogêneas para a realização de consultas e análises dimensionais.

Data Warehouse representa um importante diferencial competitivo e não é um produto e sim um ambiente.

Introdução

Uma organização moderna precisa de Sistemas de Informações eficientes e fáceis de se utilizar, a fim de sobreviver e obter sucesso em um ambiente globalizado e altamente competitivo. Neste contexto, pode-se citar várias razões para construir estes sistemas, que são:

- As decisões precisam ser tomadas rapidamente e corretamente, usando todos os dados disponíveis.
- Os usuários de sistemas de informações são especialistas de domínio de negócio, não profissionais de computação.
- O volume de dados dobra a cada 18 meses, o que afeta o tempo de resposta e, incontestavelmente, a habilidade em compreender seu conteúdo.
- A competição aumenta dia após dia nas áreas de inteligência empresarial, bem como o valor agregado de informações.

Introdução

Com a evolução dos sistemas transacionais, as empresas passam a investir nas necessidades de gerar informações executivas (business intelligence).

Existe uma grande dificuldade dos desenvolvedores de sistemas para entender a diferença entre informação e dado.

Esta afirmação é a síntese da grande diferença entre um projeto de banco de dados para sistemas transacionais (OLTP) e um projeto de banco de dados para Data Warehouse, com tecnologia OLAP.

OLTP - Online Transaction Processing

OLAP - Online Analytical Processing

Introdução

Os **dados** não possuem significado relevante e não conduzem a nenhuma compreensão. Representam algo que não tem sentido a princípio. Portanto, não podem ser usados para embasar conclusões, muito menos respaldar decisões.

A **informação** é a ordenação e organização dos dados de forma a transmitir significado e compreensão dentro de um determinado contexto. Representam o conjunto ou consolidação dos dados de forma a fundamentar o conhecimento.



Ambiente de um Data Warehouse



grande verde casa



A casa verde é grande.

Ambiente de um Data Warehouse

Vamos imaginar um exemplo de diagnóstico médico, analisando um exame de rotina:

Mede-se a temperatura a pressão arterial. Obtêm-se **dados** para uma futura análise de um especialista: o médico.

Se projetarmos um sistema para armazenar e manipular esses dados de várias formas, ainda não trataremos da informação em modo sistêmico. O que interessa saber é se o paciente: vai ter uma parada cardíaca ou um enfarto, se sua saúde está boa, etc.

Estas respostas são **informações** e podem ser projetadas através de gráficos de tendência ou cálculos estatísticos simulando possibilidades, desenvolvendo aplicações de sistema.

Para suprir esta necessidade de guardar dados para uso futuro, tem-se o surgimento do conceito de Armazém de Dados (Data Warehouse).

Ambiente de um Data Warehouse

Nos dias de hoje, não podemos imaginar um médico realizando um diagnóstico sem ao menos nos pedir um exame, caso isto ocorresse, não confiaríamos em sua opinião. Então porque devemos confiar em um empresário que toma decisões com pouca informação ou nenhuma? Não devemos pensar em fazer negócios e muito menos comprar ações ou investir nosso dinheiro em empresas em que a tomada de decisão é feita na "sorte".

Quando vamos a um médico, ficamos satisfeito quando o diagnóstico é realizado após exames detalhados de nossas condições físicas (exames de sangue, tomografias, etc.).

Similarmente, os executivos necessitam, para diagnosticar e administrar as tendências de negócio, de um ambiente que lhes permitam executar exames de seus dados com a mesma capacidade, profundidade, transparência e evolução.

Ambiente de um Data Warehouse

Quando um médico, além de seus exames, analisa seu histórico pessoal e familiar, com isto, pode diagnosticar tendências a um quadro de diabetes, problemas cardíacos, infecções respiratórias, etc.

É necessário que o executivo visualize suas tendências de vendas, por exemplo. A sazonalidade e a regionalização de vendas de determinados produtos, etc. Uma análise de dados históricos pode nos apresentar indicadores de crescimento ou sinalizadores para os negócios.

Ambiente de um Data Warehouse

Algumas diferenças entre dados de Sistemas Transacionais e Data Warehouse.

Dados Transacionais (OLTP)	Dados em um Data Warehouse (OLAP)
Baseados em aplicações.	Baseados em assuntos ou negócios.
Detalhados.	Resumidos ou refinados.
Exatos em relação do momento de acesso.	Representam valores de momentos já decorridos ou instantâneos.
Acessados uma unidade por vez.	Acessados um conjunto por vez.
Alta disponibilidade.	Disponibilidade atenuada.
Não contemplam a redundância.	A redundância não pode ser ignorada.
Estrutura fixa: conteúdos variáveis.	Estrutura flexível.
Pequena quantidade de dados usada em um processo.	Grande quantidade de dados usada em um processo.
Atendem às necessidades cotidianas.	Atendem às necessidades gerenciais.
Alta probabilidade de acesso.	Baixa, ou modesta probabilidade de acesso.

Ambiente de um Data Warehouse

Razões tecnológicas para a existência de Data Warehouse:

Primeira: o Data Warehouse é projetado para resolver a incompatibilidade de sistemas de informações transacionais e operacionais. Estas duas classes de sistemas são projetadas para satisfazer exigências diferentes, mas frequentemente incompatíveis.

Segunda: a infraestrutura (IT) muda rapidamente, do mesmo modo que suas capacidades aumentam. Isto pode ser evidenciado através dos seguintes pontos:

- O preço dos computadores que operam em uma velocidade medida em MIPS (milhões de instruções por segundo) continua caindo, enquanto o poder dos microprocessadores dobra a cada 2 anos.
- O preço de armazenamento digital está diminuindo.
- A banda passante das redes está aumentando, enquanto o preço de banda, diminuindo.
- O local de trabalho é crescentemente heterogêneo em termos de hardware e software.
- Os sistemas legados precisam, e podem, ser integrados com novas aplicações.

Referências

CARVALHO, Luis A. V. **Datamining: A mineração de dados no marketing, medicina, economia, engenharia e administração.** 1. ed. Rio de Janeiro: Ciência Moderna, 2005.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. **Data Mining: um guia prático.** Rio de Janeiro: Campus, 2005.

MACHADO, Felipe Nery Rodrigues. **Tecnologia e projeto de data warehouse.** 2. ed. São Paulo: Érica, 2006.

SILBERSCHATZ, A.; KORTH, H.; SUBARSHAN, S. **Sistema de Banco de Dados.** 5. ed. Rio de Janeiro: Campus, 2006.

TAUB, Benjamin et al; **Oracle 8i data warehouse.** Rio de Janeiro: Campus, 2001.



Aula 03

Ecossistema de informações e Sistemas Transacionais (OLTP)

Objetivo

Estudar o ciclo de vida das informações, desde o seu início até o momento que se torne obsoleto e descartado. Entender a importância dos sistemas transacionais na evolução do ambiente de banco de dados, adequado à análise de negócios apoiando a tomada de decisões gerenciais e estratégicas.

Sistemas Transacionais (OLTP)

No mundo corporativo, a informação é um dos ativos (patrimônio) mais importantes, residente na cabeça de seus empregados e muitas vezes supera o valor contábil da empresa.

Sistemas Transacionais (OLTP)

Estamos vivendo na era de "sobrecarga de informações". As companhias possuem diversos sistemas, muitas vezes com tecnologias diferentes, utilizados para automatizar e administrar seus recursos.

Estes sistemas geram e retêm uma enorme quantidade de dados estruturados de várias maneiras, além disso, as empresas também armazenam grandes volumes de dados não estruturados na forma de e-mail, documentos (doc, xls, pdf, etc.) e imagens.

As informações estruturadas e não estruturadas devem ser armazenadas e retidas de dentro destes sistemas.

Sistemas Transacionais (OLTP)

O ciclo de vida da informação passa pelas fases de:

- criação ou captura;
- armazenamento;
- versionamento;
- indexação;
- gestão;
- limpeza;
- distribuição;
- publicação;
- pesquisa;
- arquivo (ou descarte).

Sistemas Transacionais (OLTP)

Os sistemas estão provocando mudanças organizacionais e administrativas, trazendo grandes desafios para administração. Hoje, qualquer sistema que colocamos dentro da empresa deve se integrar aos já existentes e participar do sistema de informações que atende aos diversos níveis e funções organizacionais.

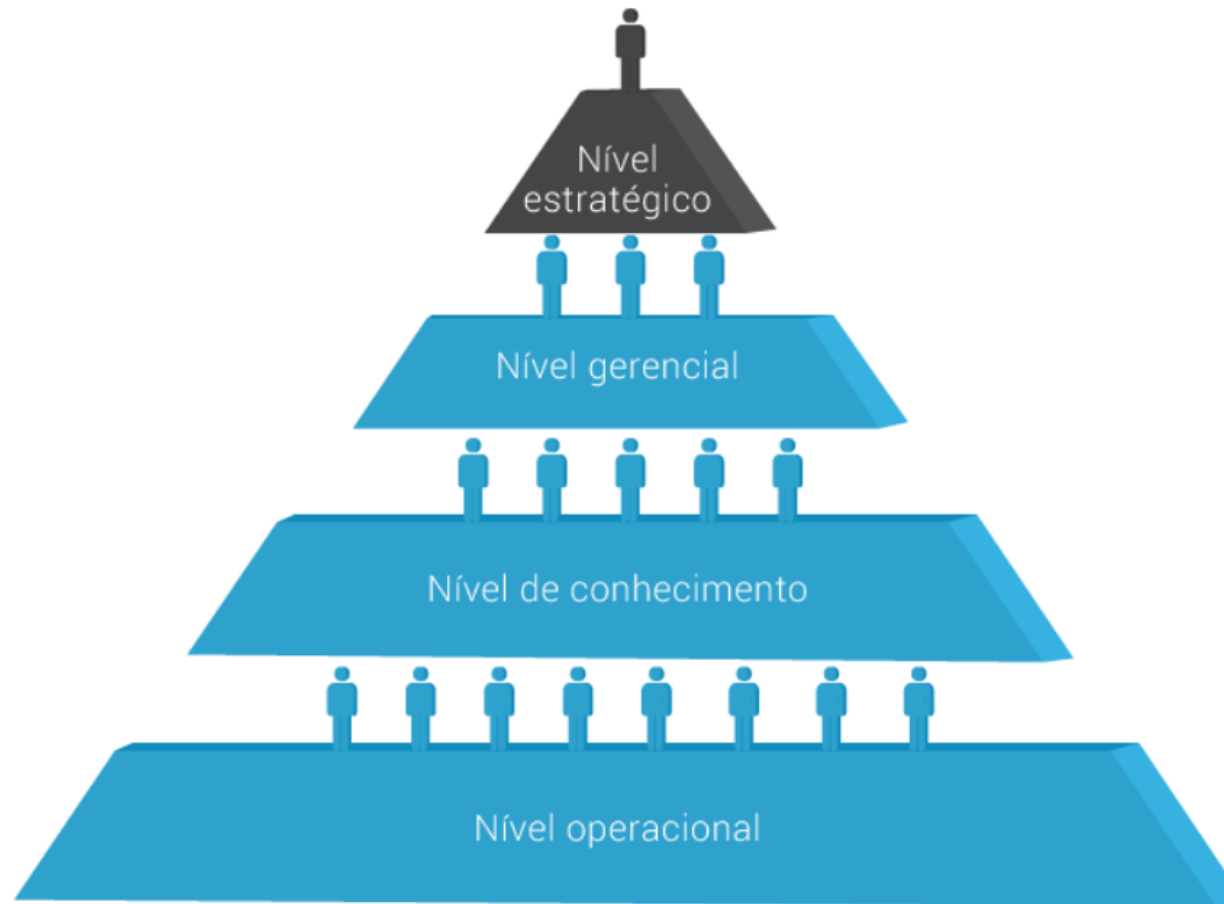
Sistemas Transacionais (OLTP)

Devido à existência de diferentes interesses e especialidades em cada nível organizacional, são necessários diversos tipos informações, pois nenhum sistema individual pode atender a todas as necessidades de uma empresa. Vamos dividir os tipos de sistemas em quatro níveis organizacionais:

- **Sistemas do nível operacional:** suporte a gerentes operacionais em transações como vendas, contas a receber, depósitos, fluxo de matéria-prima, etc.
- **Sistemas do nível de conhecimento:** envolvem as estações de trabalho e automação de escritório a fim de controlar o fluxo de documentos.
- **Sistemas do nível gerencial:** atendem a atividades de monitoração, controle, tomadas de decisões e procedimentos administrativos dos gerentes médios.
- **Sistemas de nível estratégico:** que ajudam a alta gerência a enfrentar questões e tendências, tanto no ambiente externo como interno a empresa.

Existem sistemas de informações desenvolvidos com capacidade de atender mais de um nível organizacional, mas, na maioria dos casos, desenvolvemos sistemas de informações focados em um nível organizacional e o potencializamos com integrações com outros sistemas.

Sistemas Transacionais (OLTP)



Sistemas de Processamento de Transações (Operacional)

Devem ser integrados e atendem ao nível operacional, realizando transações rotineiras, como folha de pagamento, pedidos, vendas, controle de estoque, etc. Através deles, os gerentes monitoram operações internas e externas da empresa. São críticos, pois se deixarem de funcionar, podem causar prejuízos a outras empresas e a própria.

Sistemas de Conhecimento

São de automação de escritório, atendem às necessidades do nível de conhecimento envolvendo duas classes de trabalhadores: a primeira com "formação universitária" (?!), como engenheiros, analistas financeiros, analistas administrativo etc., e a segunda com trabalhadores de dados, como secretárias, técnicos administrativos, contadores, arquivistas etc. As duas classes se diferenciam, pois trabalhadores de conhecimento criam informações e trabalhadores de dados manipulam, usam informações prontas, a produtividade dos últimos é aumentada com o uso destes sistemas, que coordenam e comunicam diversas unidades.

Sistemas de Informação Gerenciais

Têm por objetivo dar suporte ao nível gerencial através de relatórios, pesquisas nos processos correntes e nos históricos. Orientados a eventos internos, apoiando o planejamento, controle e tomada de decisão. Eles dependem dos Sistemas de Processamento de Transações para a aquisição de dados, o tratamento e a apresentação.

Sistemas de Apoio Estratégico

A maioria dos executivos das empresas têm pouca ou nenhuma experiência com computadores e são obrigados a utilizar aplicações para apoiar suas tomadas de decisões não rotineiras, que exigem bom senso avaliação e percepção. Criam um ambiente generalizado, de computação e comunicação, em vez de aplicações fixas e capacidades específicas. Projetados para incorporar dados externos como leis e novos concorrentes, também adquirem informações dos Sistemas de Informações da Empresa para apresentar de forma resumida e útil aos executivos.

Sistemas Transacionais (OLTP)

A informação tem um valor altamente significativo e pode representar grande poder para quem a possui, contém alto valor, pois está integrada com os processos, pessoas e tecnologias, é um ativo intangível como já falamos e é extremamente importante para os negócios.

A empresa passa a ter uma posição estratégica no mercado quando tem a tecnologia como aliada no controle da informação. O fator de sucesso para os negócios é dispor da informação correta, na hora adequada, isto significa tomar uma decisão de forma ágil e eficiente.

Sistemas Transacionais (OLTP)

A informação é importante tanto na definição quanto na execução de uma estratégia, pois:

- Ajuda a identificar ameaças.
- Ajuda a identificar oportunidades.
- Cria um cenário para uma resposta competitiva.
- Torna-se a matéria-prima se adotar políticas estratégicas eficazes.

Sistemas Transacionais (OLTP)

Os sistemas transacionais, conhecidos por OLTP (Online Transaction Processing), são a fonte de dados para o Data Warehouse.

A figura ao lado representa o ciclo de vida de um projeto de Data Warehouse.



Sistemas Transacionais (OLTP)

Vamos estudar a primeira fase: Fonte de Dados, que é composta pelos Sistemas Transacionais (OLTP).

OLTP são aplicações com acesso direto à informação em tempo real. Processam unidades de trabalho, chamadas de transações. Uma única transação pode solicitar uma conta bancária e atualizar o extrato para refletir um depósito.

Exemplos de aplicação: folha de pagamento, controle de vendas, controle de estoque, recursos humanos, etc.

As aplicações comerciais, geralmente, possuem muitos processos similares, por exemplo, geração de um pedido em um sistema de processamento de vendas, uma reserva de assento em um avião ou uma consulta de crédito em um sistema de controle de crédito.

A execução de um desses itens é uma transação. Isso nos leva à definir transação como uma unidade de processamento que corresponde a uma transação comercial e constitui uma entidade lógica dentro de um aplicativo.

Sistemas Transacionais (OLTP)

Em um sistema de processamento de transações, os usuários finais têm acesso online ao sistema e aos dados da empresa. Em um ambiente de transacional, muitos usuários, repetidamente, executam processos semelhantes e requerem uma resposta rápida para cada operação.

Exemplos de usuários: funcionários que emitem pedidos (balconistas), atendentes de reserva das companhias aéreas ou caixas de banco. Eles compartilham um ambiente de programas e dados em suas respectivas empresas.

Sistemas Transacionais (OLTP)

Em um típico sistema de processamento de transações, temos:

- Muitos usuários finais executam as mesmas operações, dividindo o mesmo banco de dados e arquivos.
- O sistema pode programar operações com base em atributos de prioridade.
- As operações são invocadas pela entrada de um registro, seu processamento e sua saída.
- As operações são concebidas para uma interface amigável e com tempos de resposta rápidos.
- Acesso imediato aos dados empresariais que tenham sido atualizados refletindo todas as operações anteriores.
- Mudança de dados da empresa deverá refletir imediatamente após cada operação, tal como ela é processada.

Sistemas Transacionais (OLTP)

A pré-definição de transações permite o seu controle de forma eficiente. Quando executada uma operação envolvendo um ou mais programas, é necessário garantir que no momento de sua execução não aconteça conflito entre programas. O bloqueio impede que transações possam acessar os dados enquanto o ele está sendo atualizado por outra transação

As funções básicas, que devem ser fornecidas por um sistema de processamento de transações, são:

- Processamento online.
- Alta disponibilidade.
- A resposta rápida.
- Baixo custo por transação.
- Acesso e atualização de recursos compartilhados com integridade.

Sistemas Transacionais (OLTP)

Os sistemas transacionais são base de dados para um projeto de Data Warehouse, então temos que saber qual a diferença entre esses projetos.

O Data Warehouse não substitui estes sistemas, mas os complementa.

Sistemas Transacionais X Data Warehouse

CARACTERÍSTICAS	SISTEMAS TRANSACIONAIS (OLTP)	DATA WAREHOUSE
Usuários	Operacional e Técnicos	Alta administração
Utilização	Tarefas Cotidianas	Decisões estratégicas
Uso	Operacional	Analítico
Padrão de Uso	Previsível	Difícil de prever
Número de usuários	Grande número de usuários	Poucos usuários
Unidade de trabalho	Inclusão, alteração e exclusão	Inclusão e consulta
Histórico	60 à 90 dias	5 à 10 anos
Integridade	A cada transação	A cada carga
Princípio de funcionamento	Com base em transformações	Com base em análise de dados
Valores de dados	Valores atuais e voláteis	Valores históricos e imutáveis
Detalhamento	Alto	Sumarizado
Organização dos dados	Orientado a aplicações	Orientado ao assunto

Sistemas Transacionais (OLTP)

Informações típicas de **Sistemas Transacionais**:

Quantas unidades do produto "X" o Sr. Fulano de Tal comprou?

Qual o histórico de pagamento do Sr. Fulano de Tal no último ano?

Quais as cotações realizadas esta semana que se transformaram em vendas esta semana?

Sistemas Transacionais (OLTP)

Informações que podem ser extraídas de **Sistemas Transacionais e Data Warehouse**:

Qual o volume mensal de vendas em cada estado?

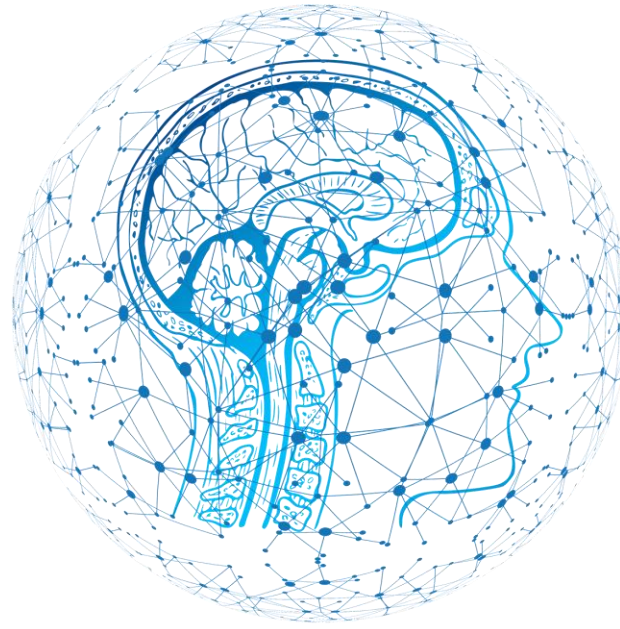
Qual o volume de inadimplência em cada filial da empresa?

Qual o percentual de efetivação de cotações para cada vendedor?

Sistemas Transacionais (OLTP)

Informações do tipo estratégico que são respondidas por um **Data Warehouse**:

- Qual é nossa posição no mercado nacional, considerando o volume de vendas do produto "X" ?
- Qual o perfil do cliente potencial para o produto "Y"?
- Aumentando as vendas em 20%, teremos mais lucros?
- Quem são meus clientes mais rentáveis?
- Quem são os melhores segmentos de clientes para um novo produto ou serviço?



Aula 04

Data warehouse e seu ciclo de vida

Objetivo

Entender a importância dos sistemas transacionais na evolução do desenvolvimento de um ambiente de banco de dados adequado à análise de negócios e ao apoio à tomada de decisões gerenciais e estratégicas.

Ciclo de vida do Data Warehouse

Tudo começa com a necessidade do negócio adquirir informações, os dados são inseridos e acessados em uma base dados regular durante todas as operações da empresa.

Ao longo do tempo, esses **dados perdem a sua importância** e são acessados com menos frequência.

Com a perda do valor comercial, os dados são **arquivados** ou **eliminados**.

Ciclo de vida do Data Warehouse

Ao entender como os dados são utilizados e por quanto tempo devem ser mantidos, as empresas podem desenvolver uma **estratégia ideal de armazenamento**, com base nos padrões de uso, **minimizando o custo total de armazenamento** durante o seu ciclo de vida.

Ciclo de vida do Data Warehouse

O Data Warehouse ou armazém de dados é o local onde é registrado a história da empresa, de seus fornecedores, de seus clientes e de suas operações de negócios, para consultas e análise.

O Data Warehouse oferece uma visão comum de dados da empresa, independentemente de como serão usados mais tarde pelos consumidores de informação (usuários do DW).

Uma vez construída uma visão comum dos dados, seus consumidores (usuários), terão a flexibilidade para analisá-los de diversas formas.

O Data Warehouse produz uma fonte estável de informação históricas, constantes, consistente e confiável para qualquer consumidor.

Ciclo de vida do Data Warehouse

As empresas têm uma enorme necessidade de informações históricas.

Por isso, o Data Warehouse pode crescer a proporções enormes (existem DWs com mais de 100 terabytes).

O projeto de DW é criado para acomodar o crescimento das informações geradas de maneira eficiente, usando sempre as regras de negócio da empresa.

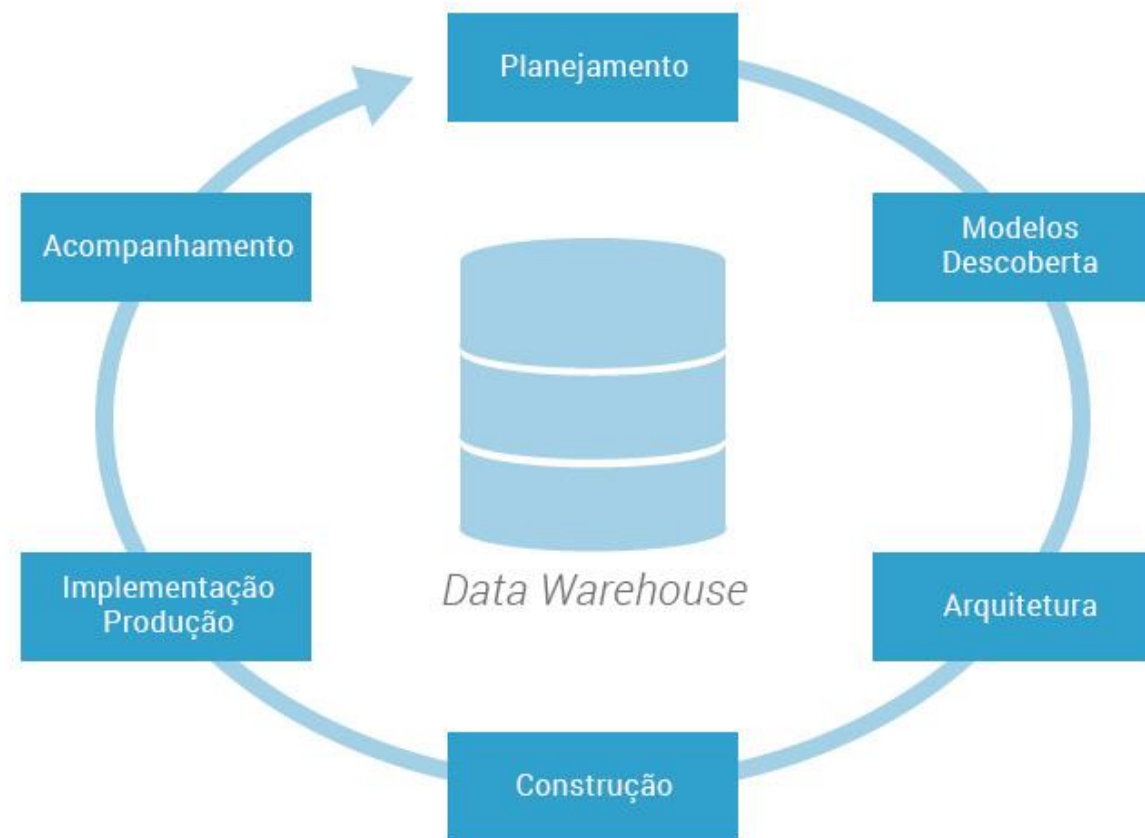
Ciclo de vida do Data Warehouse

Para facilitar as consultas no Data Warehouse, surgiram ferramentas com conceito de **OLAP (On line Analytical Processing)** que permitem realizar análise dos dados em diversas perspectivas, gerando informações gerenciais com capacidade de mergulhar na informação até atingir o seu nível mais baixo de detalhe.

Estatísticas mundiais mostram que o tempo gasto com a obtenção e análise de dados para tomada de decisão é significativamente menor quando são utilizadas aplicações de Data Warehouse, permitindo que a tomada de decisão seja mais precisa.

A crescente utilização do Data Warehouse pelas empresas está relacionada à necessidade do domínio de informações estratégicas para garantir respostas e ações rápidas, assegurando a competitividade de um mercado altamente competitivo e mutável.

Ciclo de vida do Data Warehouse



Ciclo de vida do Data Warehouse

Planejamento e visão: Definição e aprovação dos investimentos.

Modelos descoberta: Levantamento de requisitos. Nessa fase é obrigatório trabalhar junto aos usuários. O modelo de dados é um dos pilares do sucesso.

Arquitetura: Desenvolvimento de subprojetos como modelo físico, tecnologia aplicada e outros.

Construção: Criação do ETL - extração, transformação e carregamento (loading), consultas, relatórios e indicadores.

Implementação/produção: Desenvolver os planos de produção, definir a equipe de produção, os processos de manutenção e carga de dados e treinamento.

Acompanhamento: Envolve o *tunning* (ajuste) contínuo para garantir qualidade na informação com boa performance.

Ciclo de vida do Data Warehouse

Pesquise e responda:

Quais são as etapas de desenvolvimento conforme os seguintes modelos:

- cascata;
- espiral;
- fases.



Aula 05

Características de um Data Warehouse

Objetivo

Entender a importância dos sistemas transacionais na evolução do desenvolvimento de um ambiente de banco de dados adequado à análise de negócios e ao apoio à tomada de decisões gerenciais e estratégicas.

Principais características de um Data Warehouse

ORIENTADO A ASSUNTO

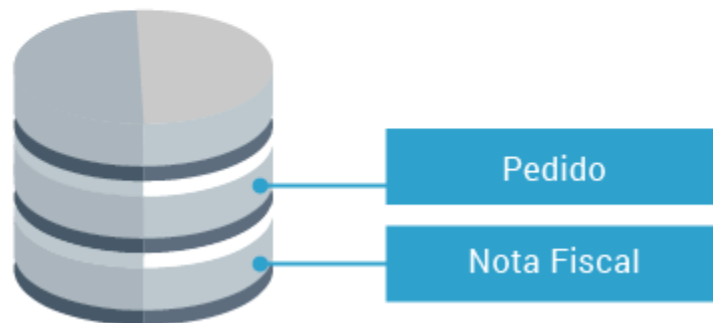
A primeira característica de um Data Warehouse é que ele está orientado ao redor do assunto principal da organização.

O percurso do dado orientado ao assunto está em contraste com a mais clássica das aplicações orientadas por processos/funções, ao redor dos quais, os sistemas operacionais estão organizados.

Principais características de um Data Warehouse

Observe o exemplo a seguir. No sistema transacional tratamos o pedido e a nota fiscal. Já no Data Warehouse tratamos o assunto vendas que possui um pedido e uma nota.

AMBIENTE TRANSACIONAL



DATA WAREHOUSE



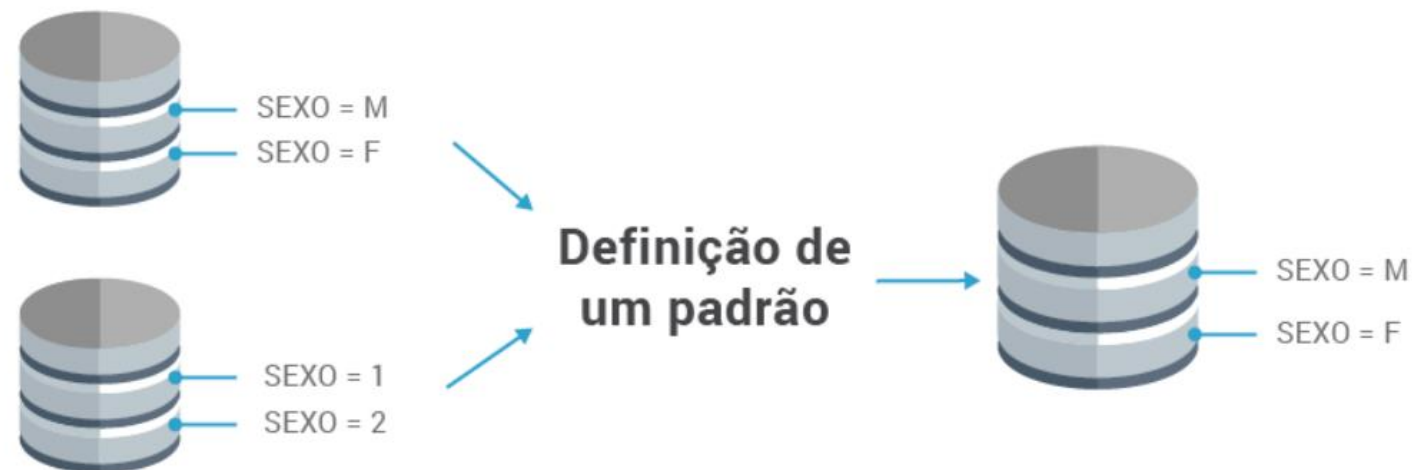
Principais características de um Data Warehouse

INTEGRADO

Os dados de um Data Warehouse possuem um alto nível de integração, isso significa que inconsistências devem ser eliminadas e que as convenções de nomes de atributo, assim como os tipos de dados, são formalmente unificados, definindo, assim, uma apresentação única para os dados do Data Warehouse, eliminando-se as possibilidades de respostas ambíguas.

Principais características de um Data Warehouse

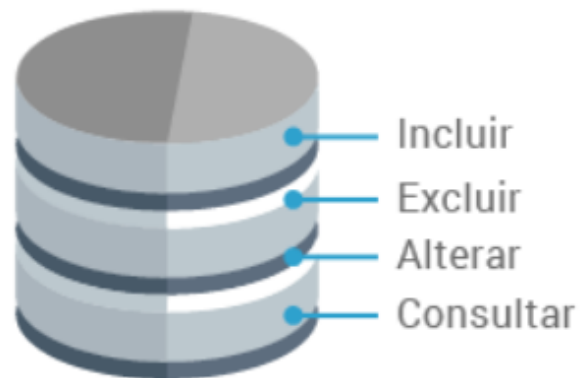
No sistema 1 o campo sexo é preenchido com M ou F, já no sistema 2 o campo sexo é preenchido com 1 ou 2. Quando levamos esta informação para um ambiente único temos que escolher como ele será preenchido e tratar o dado no momento da carga, tendo apenas um tipo de interpretação do dado. No exemplo os dados de sexo oriundos dos dois sistemas estão sendo tratados como M ou F.



Principais características de um Data Warehouse

NÃO VOLÁTIL

Um Data Warehouse possui duas operações básicas: carga de dados e acesso a eles. Estes dados se originam dos sistemas transacionais, são transformados e purificados e, em seguida, carregados no Data Warehouse. Estes dados permanecem lá até que se decida que eles não serão mais necessários.



Principais características de um Data Warehouse

Para entender melhor o processo de Data Warehouse observe a figura abaixo:



Principais características de um Data Warehouse

Em um projeto de Data Warehouse devemos repensar a forma de construir sistemas, com suas consistências entre os dados.

A construção de um Data Warehouse exige a transferência e transformação dos dados existentes em sistemas corporativos, para uma base independente.

Essa base de dados ficará disponibilizada para os usuários e mantida por meio de processos diferenciados dos existentes nos sistemas de operação transacional.



Aula 06

Levantamento de requisitos para a montagem de um Data Warehouse

Objetivo

Entender a importância da etapa de **levantamento de requisitos** para um projeto de Data Warehouse.

Introdução

Ao iniciar a construção de um Data Warehouse, a empresa tem a oportunidade para mudar a visão de processos departamentais, resultando num ciclo de informações gerenciais criadas a partir dos sistemas transacionais.

Após a análise por meio de indicadores * e relatórios, eles podem apontar a **novas oportunidades** de negócios ou para **necessidades de melhorias** nos processos já existentes.

Se realizadas, elas repercutirão na evolução dos indicadores empresariais fechando o ciclo da informação.

KPI - Key Performance Indicator (indicador-chave de desempenho)

Veja: <https://www.otimizej.com.br/conteudo/kpi-oque-e-e-importancia?>

Levantamento de requisitos para Data Warehouse

Quando realizamos um levantamento de requisitos para um determinado projeto, devemos nos atentar as percepções que estão sujeitas a **pormenores subjetivos *** ou **ruídos entre o transmissor e o receptor da mensagem**, então, devemos documentar e exemplificar todo o levantamento de requisitos.

Fundamentalmente, devemos estabelecer uma **visão comum** entre todos os membros da equipe e usuários e **eliminar esses ruídos**.

* subjetivo: baseia-se em percepções pessoais

Levantamento de requisitos para Data Warehouse

A análise de requisitos é fundamental para a concepção e desenvolvimento de um Data Warehouse.

Os requisitos solicitados pelos usuários direcionam as decisões sobre os dados a serem incorporados no banco de dados, como organizar e a frequência de atualização dos dados.

As necessidades dos usuários também determinam o tipo de ferramenta necessária e como o usuário vai interagir com os dados.

A definição de granularidade de dados não deve ser conduzida apenas pela sua disponibilidade nos sistemas fontes.

Levantamento de requisitos para Data Warehouse

Em todo projeto de Data Warehouse devemos sempre responder as seguintes questões:

- Preciso de um Data Warehouse?
- Que problemas específicos vão resolver?
- Quais são os resultados que a empresa está esperando deste projeto?
- Como os resultados serão apresentados?
- Quais os critérios que uso para medir o sucesso?
- Quais são os recursos disponíveis (tempo, dinheiro e pessoal)?
- Quais são as entradas e saídas?
- Quando o projeto começará e acabará?
- Quem é o cliente?
- Quem é o patrocinador?
- Quem são os envolvidos neste este projeto?

Levantamento de requisitos para Data Warehouse

A análise dos requisitos tem três etapas fundamentais:

- Identificar os dados e sua granularidade: os usuários devem participar ativamente desta etapa (por meio de entrevistas).
- Desenvolver uma modelagem de dados de forma que atenda a granularidade adequada ao projeto.
- Determinar de onde serão retirados os dados para alimentar o Data Warehouse.

Levantamento de requisitos para Data Warehouse

Existem outras tarefas que também devem ser abordadas como resultado do processo de análise de requisitos:

- tipos de consultas;
- ferramenta de OLAP (Online Analytical Processing);
- recursos para apoiar as interações do usuário com os dados;
- infraestrutura de hardware necessária para carregar, armazenar e fornecer dados para o usuário final;
- fonte de análise dos dados;
- estratégia de arquivamento.

Levantamento de requisitos para Data Warehouse

Os requisitos iniciais são identificados por entrevistas, com um conjunto representativo de usuários finais.

Em preparação para elas, muitas vezes é útil que os usuários finais forneçam uma amostra dos relatórios e análise elaborados em cima destes relatórios, bem como telas de todas as ferramentas que eles usam.

Levantamento de requisitos para Data Warehouse

Perguntas para entrevista com os usuários finais e com o departamento de TI:

- Quais são seus objetivos de negócio?
- Como você interpreta o conjunto de dados?
- Como devem ser organizados os dados? (por cliente, por produto, por vendedor, por região, etc.)
- Quais são as hierarquias, conjuntos ou agregados utilizados?
- Quais são as medidas ou fatos que você vai trabalhar (receitas, despesas, etc.)?
- Como você filtrará os dados?
- Quantas vezes você obtém as atualizações dos dados? (diariamente, semanalmente, mensalmente ou trimestralmente?)
- Que ferramentas você utiliza para interagir com os dados?
- Qual é a disponibilidade de dados?
- Os dados são limpos?
- Quais tipos de análises que você gostaria de fazer que não consegue fazer hoje?

Levantamento de requisitos para Data Warehouse

As entrevistas devem ser escritas como um **documento de requisitos de negócio** para identificar o **objetivo**, **finalidade** e **abrangência** do Data Warehouse.

Os requisitos de dados, bem como os requisitos funcionais, após mapeados, devem ser **validados** com os usuários finais.

Levantamento de requisitos para Data Warehouse

Alguns produtos gerados nas fases de construção de um Data Warehouse:

- **Plano de projeto:** Plano detalhado para implementação, incluindo atividades, recursos necessários, duração, precedências e produtos entregáveis.
- **Lista e modelos de documentos do produto:** Lista com definição detalhada e modelos de documentos dos produtos a serem entregues pelo projeto.
- **Especificação dos indicadores:** Especificação dos indicadores (chamados de fatos ou métricas) revisados e aprovados.
- **Requisitos do projeto:** Outros requisitos do projeto para cada etapa.
- **Modelo lógico:** Modelo lógico e dicionário de dados.
- **Mapeamento de dados:** Definição de fontes de dados, mapeamento, identificação de requerimentos de transformação com foco no modelo lógico de dados.
- **Arquitetura do sistema:** Desenho da arquitetura do sistema.
- **Modelo físico:** Desenho do modelo físico de dados correspondente, scripts para geração do banco de dados físico, criado no ambiente de desenvolvimento.

Levantamento de requisitos para Data Warehouse

Alguns produtos gerados nas fases de construção de um Data Warehouse (continuação):

- **Desenho do Front-end:** Desenho e especificação dos componentes que compõem o front-end (ferramentas OLAP).
- **Desenho de ETL:** Definição dos layouts de extração, desenho do processo de carga e transformação.
- **Planos de teste:** Planos de teste de sistemas e de aceitação.
- **Desenvolvimento do Front-end:** Relatórios, painéis, etc., conforme especificado.
- **Desenvolvimento de ETL:** Processos de carga desenvolvidos e documentados conforme definido no desenho do processo ETL e prontos para início dos testes; base de dados preparada para início dos testes.
- **Testes de sistema:** Processos de carga testados conforme plano de testes de sistema.
- **Entrega para produção:** Modelo físico de dados e rotinas de carga entregues para execução no ambiente de produção.
- **Testes de aceitação:** Scripts testados e aprovados, conforme definido no plano de testes de aceitação.

Levantamento de requisitos para Data Warehouse

Leitura recomendada:

Artigo SQL Magazine 13 - Modelagem de Data Warehouses e Data Marts – Parte 1

Disponível em: <https://www.devmedia.com.br/artigo-sql-magazine-13-modelagem-de-data-warehouses-e-data-marts-parte-1/5656>

Artigo SQL Magazine 14 - Modelagem de Data Warehouses e Data Marts – Parte 2

Disponível em: <https://www.devmedia.com.br/artigo-sql-magazine-14-modelagem-de-data-warehouses-e-data-marts-parte-ii/5686>



Aula 07

Elementos básicos da modelagem multidimensional
e granularidade de dados

Objetivo

Entender como é realizada a modelagem multidimensional e a importância da granularidade dos dados para um projeto de Data Warehouse.

Modelagem Multidimensional

Modelo multidimensional

A modelagem dimensional permite ao usuário observar seu banco de dados no formato de um **hipercubo** contendo duas, três ou quantas **dimensões** forem possíveis e aplicáveis. Esta modelagem proporciona:

- ganho de tempo na consulta;
- melhor organização do sistema;
- sua utilização de forma intuitiva para o usuário.

Modelo multidimensional

Modelos multidimensionais proporcionam uma estrutura de sistemas de informação que permitem que uma empresa:

1. disponha de acesso muito flexível a dados;
2. fatie e agrupe dados de qualquer maneira;
3. explore dinamicamente o relacionamento entre dados resumidos e detalhados.

Esses sistemas, além de oferecerem flexibilidade em relação às necessidades dos usuários, fornecem um alto nível de controle.

Modelo multidimensional

O modelo dimensional permite a visualização de dados na forma de um cubo, em que cada dimensão representa o contexto de um determinado fato, e a intersecção entre elas representa as medidas do fato.

Matematicamente, o cubo possui apenas três dimensões, entretanto, no modelo dimensional a metáfora do cubo pode possuir quantas dimensões forem necessárias para representar um determinado fato.

Um modelo multidimensional é formado por 3 elementos básicos:

1. Fatos;
2. Dimensões;
3. Medidas (variáveis).

1. Fato

A tabela **fato** está no centro do **Star schema** (esquema estrela). Ela contém as métricas de negócio, ou seja, medidas numéricas.

Cada fato representa uma transação ou um evento de negócio e é utilizado para analisar o processo de negócio de uma empresa, tudo aquilo que pode ser representado por meio de valores mensuráveis.

Fatos são objetos de análise e refletem a evolução dos negócios do dia a dia de uma organização.

Fatos podem ser caracterizados por algo que ocorre de forma variável ao longo do tempo e podem ser expressos em valores mensuráveis.

1. Fato

Como identificar um fato?

Comece por identificar o foco ou tema da análise (por exemplo, vendas, recursos humanos, finanças). A área temática deve conter as métricas descritas (por exemplo, contém o valor das vendas e unidades vendidas) e ter uma fonte de dados disponíveis (por exemplo, fonte de dados operacionais).

Dentro de cada área, identificar as transações operacionais que retratam eventos de negócios. Um exemplo de transação operacional é a de compra do cliente. Examine os dados criados por essas operações e identifique os fatos que são usados pelos processos de negócios.

Confirme com os usuários finais que você tem identificado todos os fatos que o usuário deseja saber.

2. Dimensões

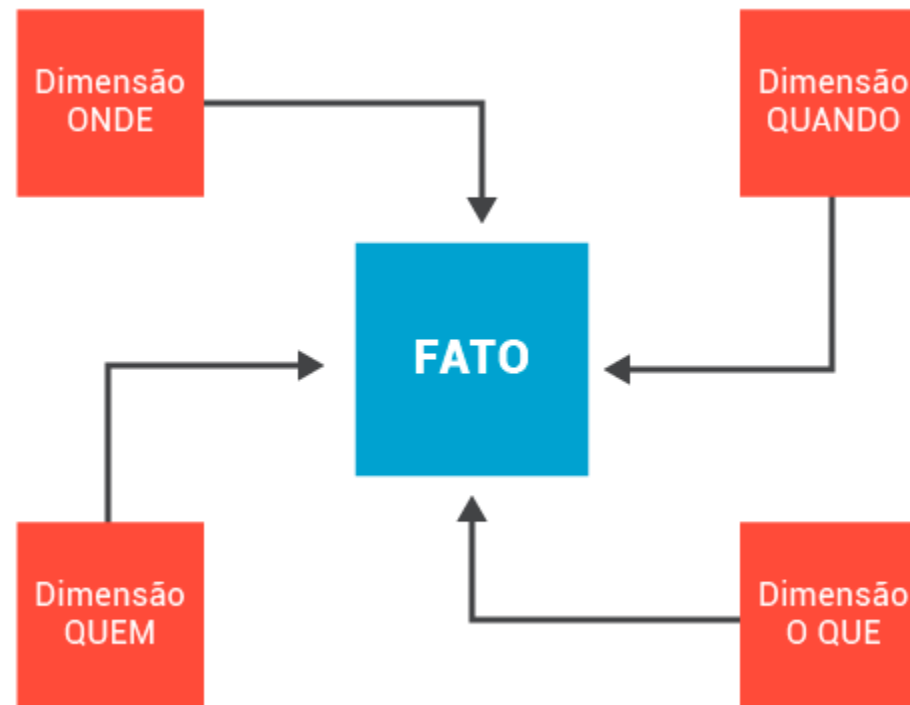
São elementos que participam de um fato, assunto de negócios. São possíveis formas de visualizar e dividir os dados, ou seja, são os "por" dos dados: "por mês", "por país", "por produto", "por região", "por funcionário", etc.

As dimensões normalmente não possuem atributos numéricos, pois são somente descritivas e classificatórias dos elementos que participam de um fato.

A maioria dos fatos envolve pelo menos quatro dimensões básicas: **onde**, **quando**, **quem** e **o que**.

A dimensão ONDE determina o local onde o fato ocorreu (local geográfico, filial) enquanto que a dimensão QUANDO é a própria dimensão tempo. Já a dimensão QUEM determina que entidades participaram do fato (cliente, fornecedor, funcionário) e a dimensão O QUÊ determina qual é o objeto do fato (produto, serviço).

2. Dimensões



2. Dimensões

Existem casos em que algumas dimensões podem conter milhões de entradas, por exemplo: a dimensão cliente em uma companhia de serviços. Neste caso, a navegação por essa dimensão pode ser lenta. Dessa forma, utilize índices nos atributos que sejam objetos de navegação.

Frequentemente, os campos mais utilizados em uma dimensão grande possuem um domínio pequeno, ou seja, assumem uma pequena quantidade de valores.

Em uma dimensão cliente, estes atributos podem ser demográficos, como sexo, faixa etária e classe social. Neste caso, pode-se optar pela criação de uma "minidimensão" separada da dimensão cliente para aumentar a eficiência da navegação.

3. Medidas (Variáveis)

São os atributos numéricos que representam um fato, o desempenho de um indicador de negócios relativo às dimensões que participam desse fato. Estes números são denominados variáveis.

3. Medidas (Variáveis)

As medidas se classificam em dois tipos:

1. **Valores aditivos:** são aqueles referentes ao fato sobre os quais podem ser aplicadas as operações de soma, subtração e média. Os valores, como: "quantidade de horas extras" e "número de alunos aprovados" representam valores aditivos.
2. **Valores não aditivos:** referentes aos fatos que não podem ser manipulados livremente, como valores percentuais ou relativos. Na realidade, eles representam os indicadores de desempenho do fato.

A tabela seguinte apresenta um exemplo de um fato com medidas ou métricas de valores aditivos e não aditivos.

os e não aditivos.

	aditivos							não aditivos	
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo	Total Semanal	Média Semanal
Quantidade de Vendas	2	5	1	4	5	3	5	25	3,57

Granularidade

Granularidade dos dados

A granularidade refere-se ao nível de detalhe ou resumo com o qual serão armazenados os dados em um Data Warehouse, quanto maior o detalhamento, mais baixo será o nível de granularidade e vice-versa. A definição da granularidade afeta diretamente o volume de dados do Data Warehouse, bem como a qualidade e desempenho das consultas a serem feitas.

Como exemplo podemos citar que uma granularidade alta garante maior rapidez nas consultas feitas, porém diminui a riqueza de informações que se pode extrair, enquanto uma menor granularidade possibilita a extração de qualquer informação, mas acarreta maior volume de dados, e, conseqüentemente, maior tempo de resposta à consulta e maior investimento em hardware.

Granularidade dos dados





Um Data Warehouse pode ser implementado em **níveis duais de granularidade** ao longo do tempo.

É possível manter as informações mais recentes em um baixo nível de granularidade, aumentando as possibilidades de extração de informações. À medida que os dados vão ficando obsoletos, é possível resumi-los em um alto nível de granularidade de forma a manter o desempenho.

O nível adequado de granularidade deve ser definido de tal forma que atenda às necessidades do usuário, tendo como limitação os recursos disponíveis, ou seja, é necessário encontrar um ponto de equilíbrio.

Granularidade dos dados

O correto é **MENOR** →

Menor agrupamento	
Granularidade	Guardar informações de cada peça. 
	Guardar informações de um conjunto de peças. 
	Existe a possibilidade de montar vários tipos de carros 
	Guarda informação do carro. 
Maior agrupamento	

O correto é **MAIOR** →

Granularidade dos dados

Sugestão de leitura:

<https://canaltech.com.br/business-intelligence/a-granularidade-de-dados-no-data-warehouse-26310/>



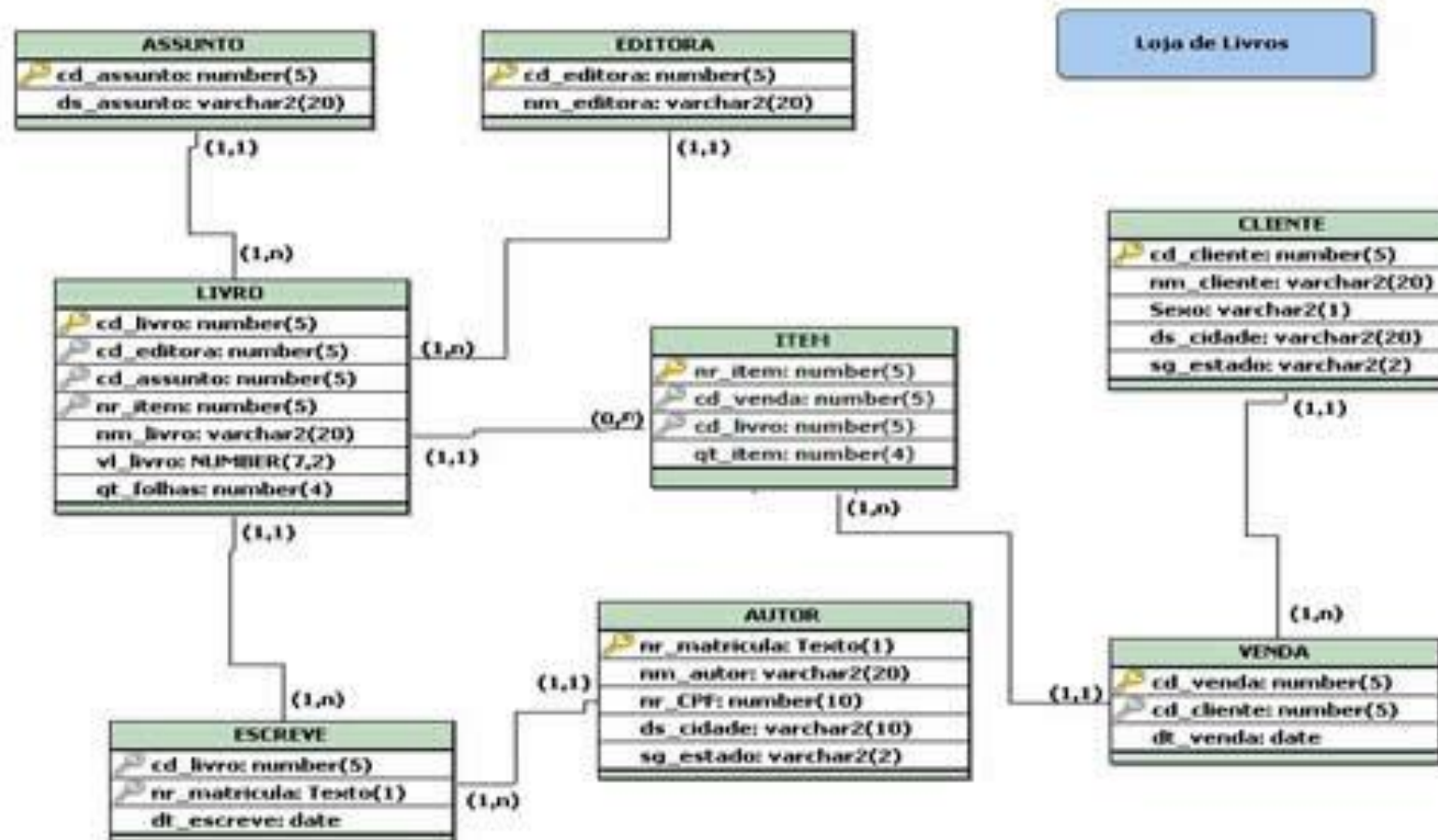
Aula 09

Criação de um modelo de Banco do Data Warehouse e um Data Mart

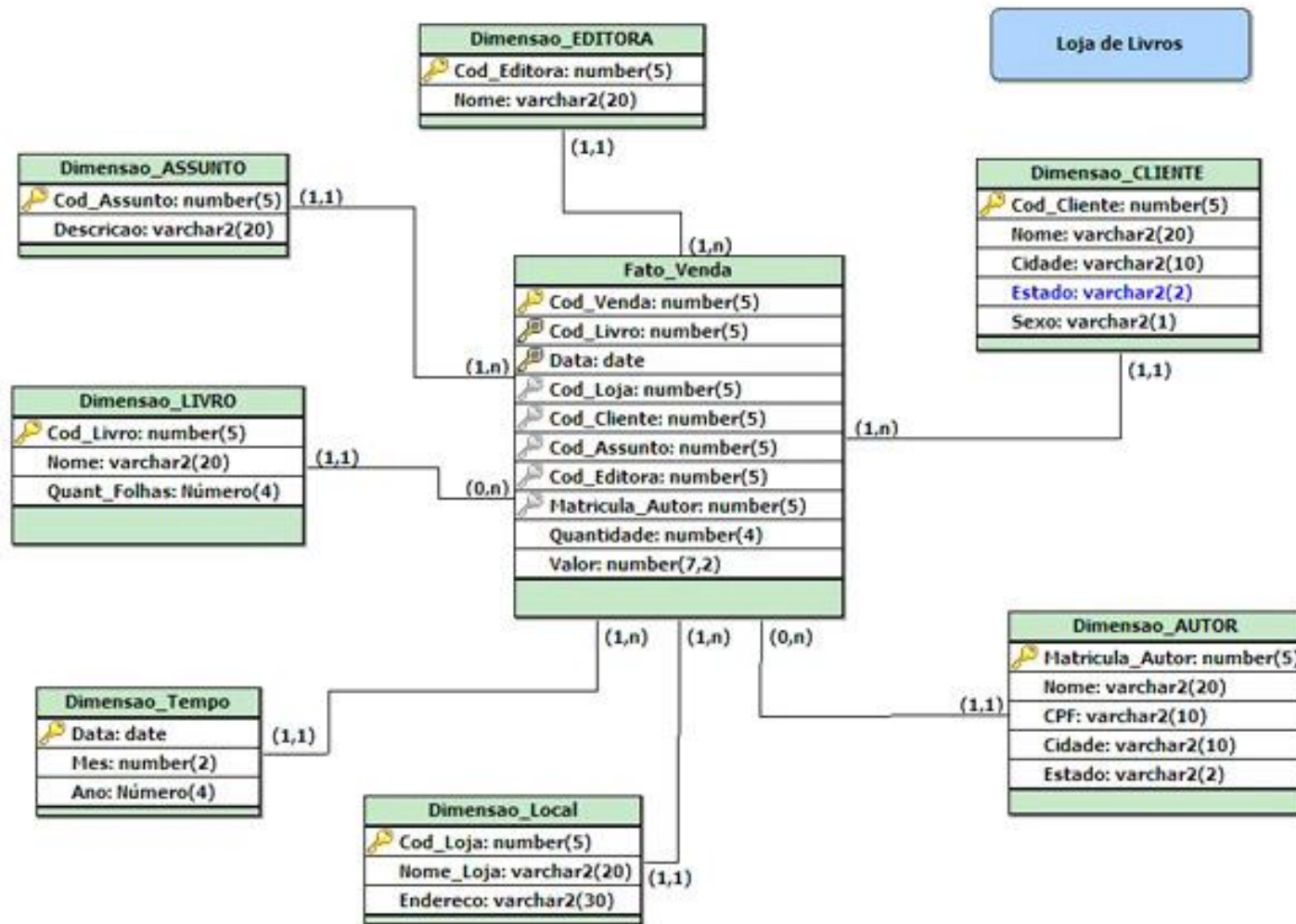
Objetivo

Entender o conceito de Data Mart e implementar o modelo Multidimensional para um Data Warehouse e um Data Mart de vendas.

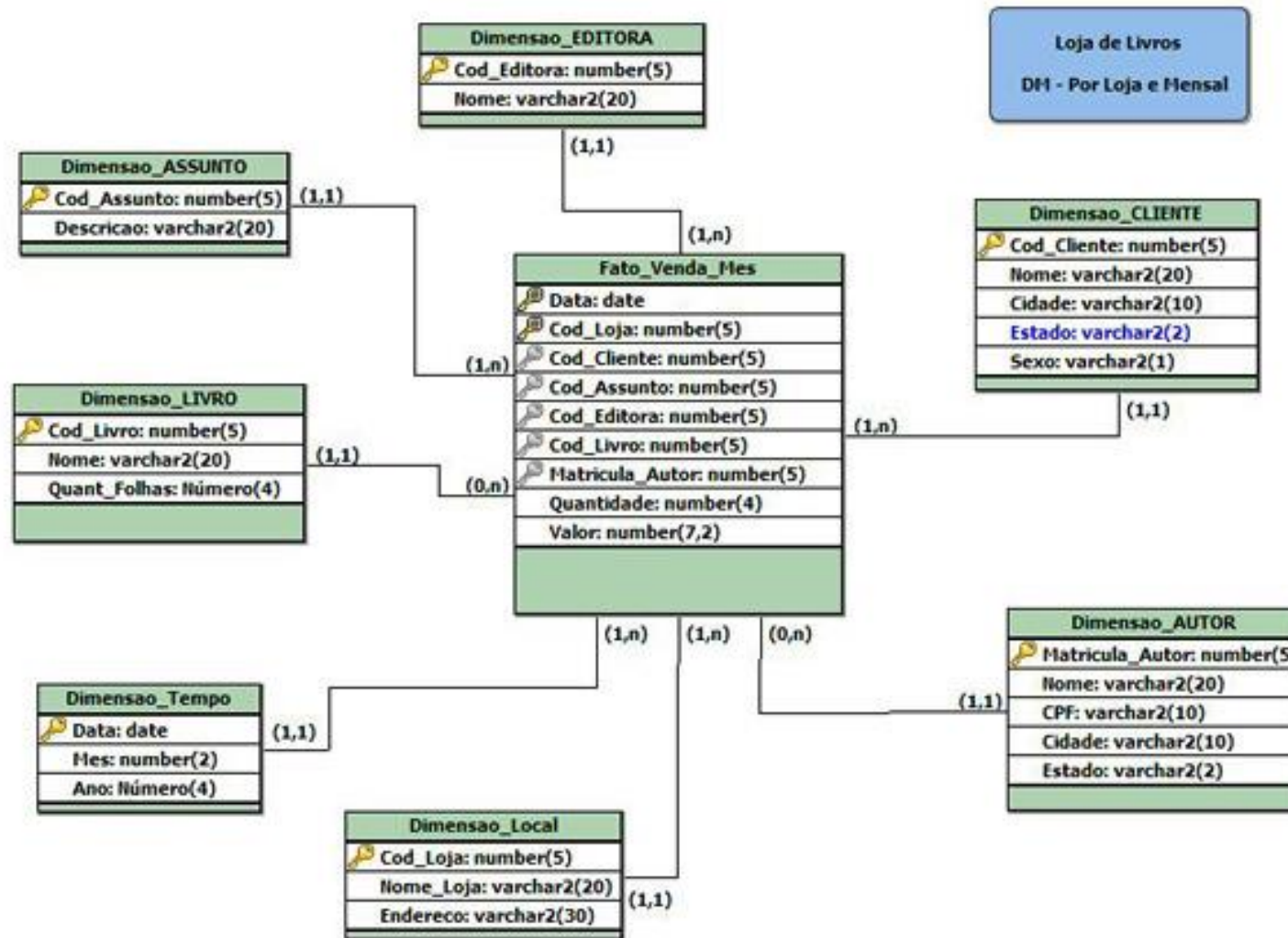
Transaccional



Multidimensional (DW)



Multidimensional (DM)

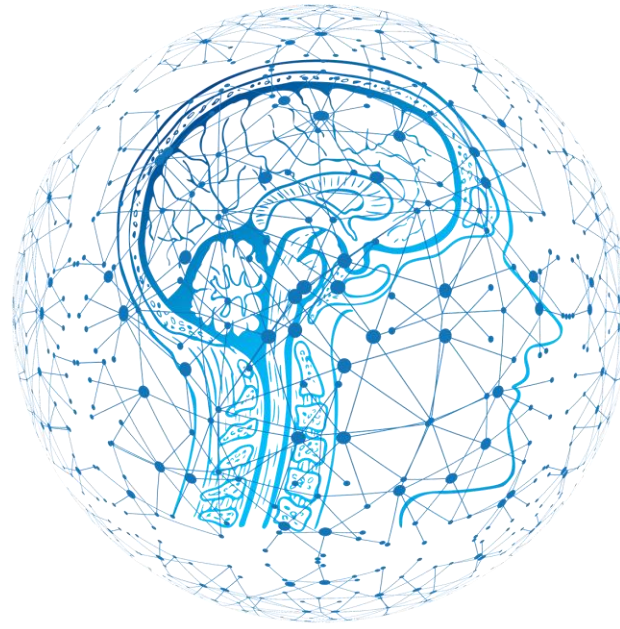


OracleXE112_Win32

Link para download:

<https://we.tl/t-pRa8LCdNdL>

Expira em: 23/10/2022



Aula 10

ETL (Extração, Limpeza, Transformação e Carga de Dados)

Objetivo

Conhecer a etapa de ETL, uma das mais críticas em um projeto de Data Warehouse.

ETL (Extract, Transform and Load)

ETL (Extract, Transform and Load), extração, transformação e carga, são os processos que permitem às empresas mover dados de múltiplas fontes, formatar, limpar e carregar em outro banco de dados, como um Data Mart ou um Data Warehouse.

As empresas sabem que têm dados valiosos espalhados ao longo de seus sistemas que precisam ser guardados em outro lugar, porque os sistemas transacionais têm como objetivo suportar os processos empresariais e não trabalhar com dados históricos. Então, temos que mover estes dados para um ambiente de Data Warehouse ou Data Mart, que nasceram com esse objetivo.

ETL (Extract, Transform and Load)

Os dados utilizados em processos de ETL podem vir de qualquer fonte: um aplicativo de mainframe, um aplicativo ERP, uma ferramenta de CRM, um arquivo simples, uma planilha do Excel, etc.

Após a extração, os dados são transformados ou modificados, para que possam ser enviados para o repositório de destino. Essas transformações dependem da lógica dos negócios envolvidos.

ETL (Extract, Transform and Load)

Extract, Transform, Load (ETL) é a tecnologia, especializada em:

Extraction: extração de dados de uma ou mais fontes;

Transformation: transformação dos dados (por exemplo, a limpeza, a reformatação, a normalização, a agregação ou a aplicação de qualquer número de regras de negócio);

Loading: carga no sistema destino do conjunto de dados, que foram extraídos dos sistemas fonte e transformados.

ETL (Extract, Transform and Load)

Extração

A extração pode ser de grandes volumes de dados, de um grande número de fontes heterogêneas, incluindo relacional, hierárquico e bancos de dados orientados a objetos, arquivos com dados estruturados ou não estruturados, documentos XML, web services e pacotes de aplicações como SAP, PeopleSoft, etc. Dados externos à organização podem ser incluídos, geralmente comprados a partir de fontes de terceiros como, pesquisas, cadastro, pontos geográficos etc.

ETL (Extract, Transform and Load)

Transformação

O segundo passo é limpar e transformar os dados. A limpeza é necessária, porque os dados normalmente provêm de uma fonte desconhecida, contendo muitas inconsistências. Isso pode ocorrer, por exemplo, se a empresa vender um determinado produto e o vendedor estiver mais preocupado em comercializá-lo do que na qualidade dos dados que está inserindo no sistema de vendas, se o cliente não informar o número de telefone na hora da compra e coloca um número qualquer para agilizar a venda como 0000-0000.

Os dados podem ter origens em diferentes sistemas e às vezes é necessário padronizar os diversos formatos. Por exemplo, o custo de um produto que é armazenado em um sistema com duas posições "0,12" e em outro é armazenado com seis posições "0,123456".

ETL (Extract, Transform and Load)

Transformação

Em alguns sistemas a informação sobre o sexo do cliente pode estar armazenada no seguinte formato "M" para masculino e "F" para feminino. Porém, em algum outro pode estar guardado como "1" para Masculino e "2" para Feminino e assim sucessivamente.

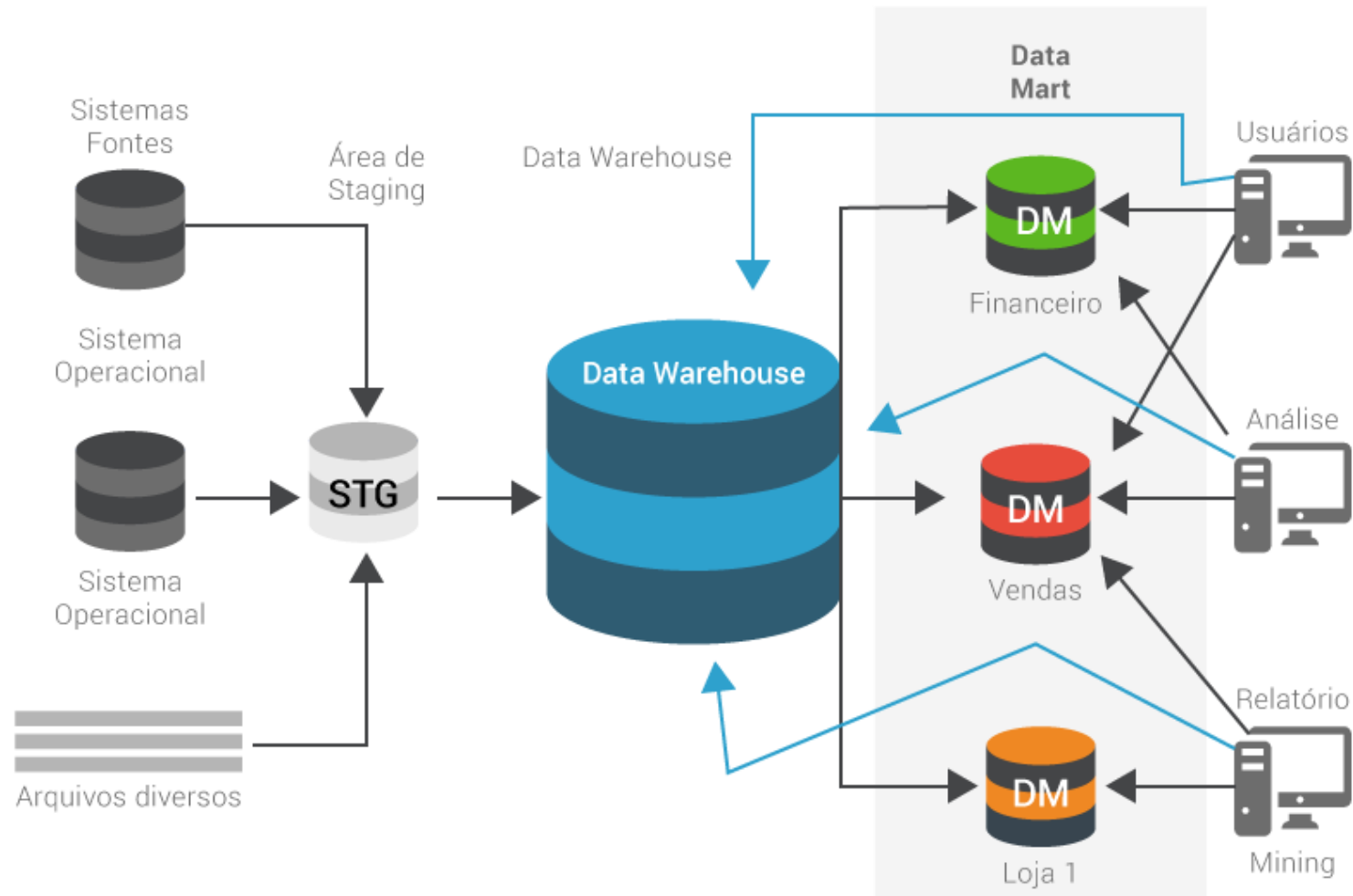
Quando levamos esses dados para o Data Warehouse, deve-se ter uma padronização, ou seja, quando o usuário for consultar as informações ele não pode ver dados iguais em formatos diferentes. Portanto, o processo de transformação tem sua importância para a uniformização dos conteúdos, em um formato normalmente sugerido pelo usuário.

ETL (Extract, Transform and Load)

Carregamento

Carregamento é a transferência de dados tratados para o aplicativo destino. Dependendo das necessidades da organização, este processo pode variar amplamente. Alguns Data Warehouses podem substituir as informações existentes semanalmente com dados acumulativos e atualizados, ao passo que outro Data Warehouse (ou até mesmo outras partes do mesmo DW) podem adicionar dados a cada hora. A temporização e o alcance de reposição, ou acréscimo, constituem opções de projetos estratégicos que dependem do tempo disponível e das necessidades dos negócios. Sistemas mais complexos podem manter um histórico e uma pista de auditoria de todas as mudanças sofridas pelos dados.

ETL (Extract, Transform and Load)



ETL (Extract, Transform and Load)

Área de Staging

Em um projeto de Data Warehouse podemos trabalhar com uma Área de Staging, que facilita o tratamento dos dados antes de inserir no banco do DW.

A área de staging representa um armazenamento intermediário, onde os dados dos sistemas de origem são copiados temporariamente. Esse process é necessário devido a vários fatores como:

- Ciclos de processamento de dados, conforme o negócio;
- Limitações de hardware;
- Recursos de rede;
- Fatores geográficos.

ETL (Extract, Transform and Load)

Área de Staging

Quando não é possível extrair todos os dados de um sistema operacional ao mesmo tempo, é necessário o armazenamento temporário nesta área.

Por exemplo, pode ser razoável extrair os dados de vendas em uma base diária, no entanto, os extratos diários podem não ser adequados para as informações financeiras, que exigem um processo de reconciliação de fim de mês. Da mesma forma, pode ser possível extrair dados de clientes a partir de um banco de dados em um país da Europa ao meio-dia, mas isso não seria viável para as informações de clientes em um sistema no Brasil.

A área de staging também é utilizada para facilitar os programas de limpeza e transformação dos dados antes da atualização do Data Warehouse.

ETL (Extract, Transform and Load)

Data Marts

Os Data Marts são bancos de dados modelados multidimensionalmente, orientados aos departamentos de uma empresa como, a área comercial, a financeira e localidades. Podemos construir vários Data Marts que depois passam a alimentar um Data Warehouse ou a fazer o caminho inverso, alimentar os Data Marts. Então, com essa fundamentação, podemos definir que os DM's podem surgir de duas maneiras. A primeira é top-down e a segunda é a botton-up.

Para criação dos Data Marts existe a presença de ETL a fim de realizar a extração, tratamento e carga dos dados.

ETL (Extract, Transform and Load)

PL-SQL: Procedural Linguagem - Structured Query Language

Para entender os processos de ETL (Extract, Transform and Load), vamos construir programas PL-SQL para carregar os dados do banco Transacional para o Dimensional.

PL/SQL: Procedural Language - Structured Query Language

Principais recursos da linguagem:

- Executar comandos SQL para manipular dados nas tabelas
- Criar constantes e variáveis
- Criar cursores para tratar o resultado de uma consulta
- Criar registros para guarda o resultado de um cursor ou campo de tabela
- Tratar erros
- Utilizar comandos de controle (if, if-then-else, case) e repetição (loop, for, while)

PL/SQL: Procedural Language - Structured Query Language

Vantagens:

- Versatilidade
- Portabilidade
- Integração com o SGBD (Sistema Gerenciador de Banco de Dados)
- Capacidade procedural (comandos de controle e repetição)
- Redução de tráfego de rede

Estrutura de um bloco PL/SQL

Estrutura de um bloco PL/SQL anônimo:

DECLARE

Inicializações, declaração de constantes, variáveis e cursores

BEGIN

Comandos SQL, estruturas de programação e outros blocos PL/SQL

BEGIN

...

END;

EXCEPTION (opcional)

Tratamento de exceções, emissão de mensagens

END;

Nota: Além de blocos anônimos, a PL/SQL permite a criação de PROCEDURES, FUNCTIONS, TRIGGERS, etc. que serão abordados neste curso.

Declarações

Na área DECLARE podemos declarar:

- constantes
- variáveis
- cursores
- estruturas
- tabelas

Tipos de Dados

Os principais tipos de dados simples que podem ser utilizados em declarações PL/SQL são:

- **CHAR:** Alfanumérico, tamanho fixo, limite: 2000 caracteres
- **VARCHAR2:** Alfanumérico, tamanho variável, limite: 4000 caracteres
- **CLOB:** (Character Long Object) Alfanumérico, tamanho variável, limite 4 Gb
- **LONG:** Alfanumérico, limites: 2 GB, apenas um por tabela
- **ROWID:** Armazena os valores dos ROWIDs das linhas das tabelas
- **BLOB:** (Binary Long Object) Binário, tamanho variável, limite: 4 Gb
- **BFILE:** (Binary File) Armazena uma referência a um arquivo externo
- **RAW:** Hexadecimais, tamanho variável, limite: 2 Kb
- **LONG ROW:** Hexadecimais, tamanho variável, limite: 2 Gb
- **NUMBER:** Numérico, limite: 38 dígitos
 - SUBTIPOS: **DECIMAL, INT, NUMERIC, FLOAT**
- **DATE:** Data e hora
- **TIMESTAMP:** Data e hora (com milésimos de segundo)
- **BOOLEAN:** Armazena os valores: TRUE, FALSE ou NULL

Constantes

Para declarações de constantes, a palavra **CONSTANT** deve aparecer antes do tipo de dado e a seguir utiliza-se o operador **:=** para atribuir-lhe um valor.

Exemplo:

```
DECLARE
```

```
    PI CONSTANT NUMBER(3,2) := 3.14;
```

```
    ...
```

```
BEGIN
```

```
    ...
```

```
END;
```

```
/
```


Variáveis

As variáveis são inicializadas de maneira similar às constantes: declarando-se o tipo e atribuindo-lhe um valor.

Variáveis não inicializadas explicitamente recebem o valor NULL.

Pode-se aplicar a restrição NOT NULL a uma variável. Neste caso, ela deverá ser inicializada.

Exemplos:

```
v1 NUMBER(4) := 1;
```

```
v2 NUMBER(4) NOT NULL:= 1;
```

Variáveis

Podemos atribuir valores às variáveis de duas maneiras:

- Utilizando o operador de atribuição:

```
total := quant * valor;
```

- Utilizando um comando SELECT com a cláusula INTO:

```
SELECT id_cliente, nome_cliente  
INTO v_id, v_nome  
FROM cliente;
```

Herança de Tipo e Tamanho

As constantes e variáveis podem herdar o tipo de outras variáveis, de colunas ou até da linha inteira de uma tabela.

Desta forma, diminuem-se as manutenções oriundas de alterações realizadas nas colunas de tabelas (Exemplo: Quando altera-se o tamanho da coluna).

Herdando o tipo de uma variável previamente declarada:

```
v_nome v_nome_cliente%type;
```

Herdando o tipo de uma coluna de uma tabela:

```
v_nome cliente.nome%type;
```

Herdando o tipo de uma linha inteira de uma tabela:

```
v_cliente cliente%rowtype;
```

Comandos SQL dentro de uma bloco PL/SQL

Comandos DML (SELECT, INSERT, UPDATE e DELETE) podem ser utilizados dentro de um bloco PL/SQL.

O comando SELECT deverá receber obrigatoriamente a cláusula INTO para que o resultado seja armazenado em variáveis e deverá retornar apenas uma linha.

Caso mais de uma linha seja retornada, apresentará o erro: **too_many_rows** e se não retornar nenhuma linha, apresentará o erro: **no_data_found**. (Detalhes serão apresentados em Tratamento de Exceções).

```
CREATE TABLE CLIENTE (  
  ID NUMBER(5),  
  NOME VARCHAR2(40));
```

```
INSERT INTO CLIENTE (ID,NOME) VALUES (1, 'ANTONIO');  
INSERT INTO CLIENTE (ID,NOME) VALUES (2, 'BEATRIZ');
```

```
DECLARE  
  V_RA ALUNO.RA%TYPE;  
  V_NOME ALUNO.NOME%TYPE;  
BEGIN  
  SELECT ID, NOME INTO V_ID, V_NOME FROM CLIENTE WHERE ID=1;  
  DBMS_OUTPUT.PUT_LINE(V_ID || ' - ' || V_NOME);  
END;  
/
```

Na PL/SQL || é usado para concatenação.

Instruções IF-THEN-ELSE e CASE

IF-THEN-ELSE

Executa um conjunto de ações de acordo com uma ou mais condições.

DECLARE

```
V_RA ALUNO.RA%TYPE := 1001;
```

```
V_NOTA ALUNO.NOTA%TYPE;
```

```
V_CONCEITO VARCHAR2(12);
```

BEGIN

```
SELECT NOTA INTO V_NOTA FROM ALUNO WHERE RA = V_RA;
```

```
IF V_NOTA <= 5 THEN V_CONCEITO := 'REGULAR';
```

```
ELSIF V_NOTA < 7 THEN V_CONCEITO := 'BOM';
```

```
ELSE V_CONCEITO := 'EXCELENTE';
```

```
END IF;
```

```
DBMS_OUTPUT.PUT_LINE (V_CONCEITO);
```

```
END;
```

```
/
```

Instruções IF-THEN-ELSE e CASE

CASE

Retorna determinado resultado de acordo com o valor da variável de comparação.

DECLARE

```
V_RA ALUNO.RA%TYPE := 1;
```

```
V_NOTA ALUNO.NOTA%TYPE;
```

```
V_CONCEITO VARCHAR2(12);
```

BEGIN

```
SELECT NOTA INTO V_NOTA FROM ALUNO WHERE RA = V_RA;
```

```
V_CONCEITO :=
```

```
CASE
```

```
  WHEN V_NOTA <= 5 THEN 'REGULAR'
```

```
  WHEN V_NOTA < 7 THEN 'BOM'
```

```
  ELSE 'EXCELENTE'
```

```
END;
```

```
DBMS_OUTPUT.PUT_LINE (V_CONCEITO);
```

```
END;
```

```
/
```

Instruções LOOP, FOR e WHILE

LOOP

Repete um bloco de comando n vezes, até que a variável contadora atinja o seu valor final.

A variável contadora não deve ser declarada na seção DECLARE e deixará de existir após a execução do comando END LOOP.

DECLARE

```
V_AUX NUMBER(2) := 0;
```

BEGIN

```
FOR V_CONTADOR IN 1..10
```

LOOP

```
V_AUX := V_AUX +1;
```

```
DBMS_OUTPUT.PUT_LINE (V_AUX);
```

```
END LOOP;
```

```
END;
```

```
/
```

Instruções LOOP, FOR e WHILE

WHILE

Repete um bloco de comandos enquanto a condição que segue o comando WHILE for verdadeira.

```
DECLARE
```

```
  V_AUX NUMBER(2) := 0;
```

```
BEGIN
```

```
  WHILE V_AUX < 10
```

```
  LOOP
```

```
    V_AUX := V_AUX +1;
```

```
    DBMS_OUTPUT.PUT_LINE (V_AUX) ;
```

```
  END LOOP;
```

```
END;
```

```
/
```


Tratamento de Exceções

Exceções são erros ou imprevistos que podem ocorrer durante a execução de um bloco PL/SQL.

Nesses casos, o gerenciador de banco de dados aborta a execução e procura a área de exceções **EXCEPTIONS**.

As exceções podem ser:

- Predefinidas
- Definidas pelo usuário

Tratamento de Exceções

PREDEFINIDAS

Disparadas automaticamente quando, no bloco PL/SQL, uma regra Oracle for violada. Podem ser identificadas por um número ou um nome.

ERRO	NOME	DESCRIÇÃO
ORA-00001	DUP_VAL_ON_INDEX	Tentativa de armazenar valor duplicado em uma coluna que possui chave primária ou única.
ORA-01012	NOT_LOGGED_ON	Tentativa acessar o banco de dados sem estar conectado a ele.
ORA-01403	NO_DATA_FOUND	Ocorre quando um comando SELECT ... INTO não retorna nenhuma linha.
ORA-01422	TOO_MANY_ROWS	Ocorre quando um comando SELECT ... INTO retorna mais de uma linha.
ORA-01476	ZERO_DIVIDE	Tentativa de dividir qualquer número por zero.

Tratamento de Exceções

PREDEFINIDAS

Exemplo:

```
DECLARE
  V_RA ALUNO.RA%TYPE;
  V_NOME ALUNO.NOME%TYPE;
BEGIN
  SELECT RA, NOME INTO V_RA, V_NOME FROM ALUNO WHERE RA=1001;
  DBMS_OUTPUT.PUT_LINE(V_RA || ' - ' || V_NOME);
EXCEPTION
  WHEN NO_DATA_FOUND
  THEN DBMS_OUTPUT.PUT_LINE ('Não há nenhum aluno com este RA');
  WHEN TOO_MANY_ROWS
  THEN DBMS_OUTPUT.PUT_LINE ('Há mais de um aluno com este RA');
  WHEN OTHERS
  THEN DBMS_OUTPUT.PUT_LINE ('Erro desconhecido');
END;
```

Tratamento de Exceções

DEFINIDAS PELO USUÁRIO

Além dos erros tratados automaticamente pelo Oracle, regras de negócio específicas podem ser tratadas. As exceções definidas pelo usuário devem ser declaradas e chamadas explicitamente pelo comando RAISE.

```
DECLARE
  V_RA ALUNO.RA%TYPE;
  V_NOTA ALUNO.NOTA%TYPE;
  V_CONTA NUMBER(2);
  ALUNO_EXCEPT EXCEPTION;
BEGIN
  SELECT COUNT(RA) INTO V_CONTA FROM ALUNO;
  IF V_CONTA = 10 THEN
    RAISE ALUNO_EXCEPT;
  ELSE INSERT INTO ALUNO VALUES (20,'SILVA');
  END IF;
EXCEPTION
  WHEN CONTA_ALUNO THEN DBMS_OUTPUT.PUT_LINE('Turma cheia!');
END;
/
```

Cursores

Cursores são áreas compostas de linhas e colunas em memória que servem para armazenar o resultado de uma seleção que retorna 0 (zero) ou mais linhas.

Na PL/SQL os cursores podem ser de dois tipos:

- Explícitos
- Implícitos

Cursosores

EXPLÍCITOS

São utilizados para execução de consultas que possam retornar nenhuma ou mais de uma linha.

Este tipo de cursor deve ser explicitamente declarado na área DECLARE.

Após sua declaração, o cursor deverá ser manipulado com o uso de alguns comandos:

- **OPEN:** abre o cursor
- **FETCH:** disponibiliza a linha corrente e posiciona na próxima linha do cursor
- **CLOSE:** fecha o cursor

Cursors

EXPLÍCITOS

Para cada cursor, quatro atributos podem ser verificados, e seus valores podem ser alterados a cada execução de um comando FETCH. Esses atributos são:

- **nome_do_cursor%FOUND:** Retorna TRUE caso FETCH consiga retornar alguma linha e FALSE caso contrário.
- **nome_do_cursor%NOTFOUND:** Retorna FALSE caso FETCH consiga retornar alguma linha e TRUE caso contrário.
- **nome_do_cursor%ROWCOUNT:** Retorna o número de linhas já processadas pelo cursor.
- **nome_do_cursor%ISOPEN:** Retorna TRUE caso o cursor esteja aberto e FALSE caso contrário.

Cursores

EXPLÍCITOS

Exemplo:

```
DECLARE
```

```
    CURSOR C_CLIENTE IS SELECT ID, NOME FROM CLIENTE;
```

```
    V_CLIENTE C_CLIENTE%ROWTYPE;
```

```
BEGIN
```

```
    OPEN C_CLIENTE;
```

```
    LOOP
```

```
        FETCH C_CLIENTE INTO V_CLIENTE;
```

```
        EXIT WHEN C_CLIENTE%NOTFOUND;
```

```
        DBMS_OUTPUT.PUT_LINE(V_CLIENTE.NOME) ;
```

```
    END LOOP;
```

```
    CLOSE C_CLIENTE;
```

```
END;
```

```
/
```


Cursosores

IMPLÍCITOS

Utilizados para processar instruções (**SELECT ... INTO**, **INSERT**, **UPDATE** e **DELETE**).

Os comandos **OPEN**, **FETCH** e **CLOSE** não podem ser aplicados a este tipo de cursor.

Cursosores implícitos esperam que apenas uma linha seja retornada. Por isso exceções tais como **NO_DATA_FOUND** (nenhuma linha satisfaz os critérios de seleção) ou **TOO_MANY_ROWS** (mais de uma linha satisfaz o critério de seleção) devem ser observadas.

Os atributos **%FOUND**, **%NOTFOUND**, **%ROWCOUNT** e **%ISOPEN** também podem ser verificados neste tipo de cursor.

Functions

Subprogramas que executam uma determinada ação e retornam valores. Portanto, podem ser invocadas por meio de um comando SELECT.

```
CREATE OR REPLACE FUNCTION SOMA (P_1 NUMBER,P_2 NUMBER)
  RETURN NUMBER
  IS
    V_TOTAL NUMBER;
BEGIN
    V_TOTAL := P_1 + P_2;
    RETURN V_TOTAL;
END SOMA;
/

SELECT SOMA (2,5) FROM DUAL;
```

Procedures

Subprogramas que executam uma determinada ação. Não retornam valores diretamente para o ambiente. Portanto, não podem ser utilizadas como argumento em um comando SELECT.

```
CREATE OR REPLACE PROCEDURE SOMA (  
    P_1 NUMBER,  
    P_2 NUMBER) IS  
    V_TOTAL NUMBER;  
BEGIN  
    V_TOTAL := P_1 + P_2;  
    DBMS_OUTPUT.PUT_LINE(V_TOTAL) ;  
END SOMA;  
/
```

```
SET SERVEROUTPUT ON
```

```
EXEC SOMA (2, 5);
```

NOTA: O comando **SET SERVEROUTPUT ON** é usado para visualizar a saída (resultado) acima.

Triggers

Blocos PL/SQL disparados automática e implicitamente sempre que ocorrer um evento associado a uma tabela (INSERT, UPDATE ou DELETE).

```
CREATE TABLE PRODUTO (  
  CODIGO NUMBER (4),  
  VALOR NUMBER (7,2) );
```

```
CREATE TABLE HISTORICO (  
  CODIGO NUMBER (4),  
  VALOR_ANTIGO NUMBER (7,2),  
  VALOR_NOVO NUMBER (7,2),  
  DATA_ATUAL DATE );
```

```
CREATE OR REPLACE TRIGGER VALOR_HISTORICO  
  BEFORE UPDATE OF VALOR ON PRODUTO  
  FOR EACH ROW  
BEGIN  
  INSERT INTO HISTORICO VALUES (:OLD.CODIGO, :OLD.VALOR, :NEW.VALOR, SYSDATE);  
END;  
/
```

Packages

Objetos do banco de dados equivalentes a bibliotecas que armazenam:

- procedures
- functions
- definições de cursores
- variáveis e constantes
- definições de exceções

Um package é composto de duas partes:

1. **Especificação:** Área onde são feitas as declarações públicas. As variáveis, constantes, cursores, exceções e subprogramas estarão disponíveis para uso externo ao package.
2. **Corpo:** Área onde são feitas as declarações privadas que estarão disponíveis apenas dentro do package e a definição de ações para os subprogramas públicos e privados.

O Corpo do package é um objeto de dicionário de dados separado da Especificação. Ele não poderá ser compilado com sucesso a menos que a Especificação já tenha sido compilada.

Packages

Exemplo:

```
CREATE OR REPLACE PACKAGE PACK_1 IS
    PROCEDURE PROC_1;
    FUNCTION FUNC_1 RETURN VARCHAR2;
END PACK_1;
/

CREATE OR REPLACE PACKAGE BODY PACK_1 IS
    PROCEDURE PROC_1
    IS
    BEGIN
        DBMS_OUTPUT.PUT_LINE('Mensagem da Procedure');
    END PROC_1;
    FUNCTION FUNC_1 RETURN VARCHAR2 IS
    BEGIN
        RETURN('Mensagem da Function');
    END FUNC_1;
END PACK_1;
/

EXEC PACK_1.PROC_1;

SELECT PACK_1.FUNC_1 FROM DUAL;
```