

MBA em Gestão da Informação e *Business Intelligence*

Arquitetura de Data Warehouse

Ricardo Holanda, MSc.
Ago/2014



Curriculum do Professor

- Doutorando em Engenharia de Teleinformática pela UFC. Mestre em Computação e Especialista em Direito Empresarial pela UECE. Graduado em Ciências da Computação pela UECE e em Direito pela UFC.
- Atua com pesquisa ligadas a PolSAR, Parallelismo, Otimização, Sistemas De Apoio a Decisão e Sistemas Especialistas.
- Atuou como Gerente de TIC da CAMED e como Coordenador de Desenvolvimento da Secretaria da Justiça e Cidadania do Estado do Ceará, desenvolvendo trabalhos de identificação biométrica, Business Intelligence, Data Warehouse dentre outros.
- Atualmente é Analista de Sistema do SERPRO, desenvolvendo trabalhos ligados a Data Warehouse e a Administração de Bancos de Dados, e Professor da UECE, FATENE e FATE.
- Possui mais de 17 anos de experiência em informática;



Ementa da Disciplina

- Conceitos de Data Warehouse. Arquitetura do Data Warehouse, modelagem de dados multidimensional, desenvolvimento do Data Warehouse, Data Martas e Ferramentas de OLAP.



Objetivos do Curso

- Apresentar conceitos e tecnologias utilizadas no desenvolvimento de um Data Warehouse de forma a aplicá-lo na análise de informações visando à decisões em empresas.



**PUC
GOIÁS**



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Conteúdo Programático

UNIDADE I: FUNDAMENTOS DE DATA WAREHOUSE

- Definição, Características e Estrutura;
- Sistemas de Apoio à Decisão;
- Executive Information Systems;
- Data Warehouse e Data Marts;
- Visão Geral dos Componentes de um DW.

UNIDADE II: COMO CONSTRUIR UM DATA WAREHOUSE

- O Ambiente de Data Warehouse;
- Projeto de um Data Warehouse;
- Definição dos Requisitos de Negócio para um DW;
- Granularidade no Data Warehouse;
- Data Warehouse Distribuído.

Conteúdo Programático

UNIDADE III: ARQUITETURA E INFRAESTRUTURA DE DW

- Arquitetura de um DW: Componentes e suas características;
- Metadados: Gerência, Armazenamento e Integração.

UNIDADE IV: PROJETO DE DADOS EM UM DW

- Princípios de modelagem dimensional e multidimensional;
- Esquemas Star e Snowflakes;
- Extração de dados, transformação e carga(ETL);
- Qualidade dos dados armazenados;
- Ferramentas e Operações OLAP;
- Modelo OLAP e variações: ROLAP e MOLAP.

Critério de Avaliação

- Trabalho em Grupo
 - Entre 3 a 5 pessoas;
 - Composição:
 - Escrito (formato acadêmico): Normas ABNT;
 - Apresentação: Todos devem participar;
 - Data da entrega do trabalho escrito e da apresentação: 28/09/2014;
 - Sorteio dos temas: 31/08/2014.
- **Não será admitido cópia de trabalhos ou textos de outros autores sem que sejam seguidas as regras da ABNT;**



Questões Gerais

- Presença em 75% das aulas:
 - Chamada obrigatória;
- Entrega/apresentação do trabalhos:
 - A nota será parcial 50% para o trabalho escrito e 50% para a apresentação;
 - Não haverá substituição do trabalho por qualquer outra atividade;
 - O trabalho escrito deverá ser entregue encadernado, junto com CD contendo a versão eletrônica editável do trabalho e da apresentação;
 - Entregas realizadas em sala de aula; e-mail não serão aceitos.



Temas

- 1. Business Intelligence: Open Souce ou Software Proprietário?**
- 2. Pentaho: Uma avaliação da Suite open source sob o ponto de vista da usabilidade;**
- 3. Data Mining: Comparativo das principais soluções open source;**
- 4. Fatores de sucesso na construção de Data Warehouse;**
- 5. Big Data e seus impactos na construção de Data Warehouse;**
- 6. Impactos da Cloud Computing em Data Warehouse;**



Referências

Bibliografia Básica (títulos , periódicos, etc.)

- INMON, William H.; WELCH, J. D.; GLASSEY, Katherine L. **Gerenciando o Data Warehouse.** São Paulo: Makron, 1999.
- INMON, William H. **Como Construir o Data Warehouse.** Rio de Janeiro: Campus, 1997. v1.
- KIMBALL, Ralph. **Data Warehouse Toolkit: Técnicas para Construção de Data Warehouses Dimensionais.** Rio de Janeiro: Campus, 1998.
- MACHADO, Felipe Nery Rodrigues. **Tecnologia E Projeto De Data Warehouse.** Ed. Erica, 2004.



Dúvidas?



Business School Brasil



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Fundamentos de Data Warehouse

Contexto:

- No mundo cada vez mais globalizado e competitivo as empresas procuram:
 - Tomar decisões mais rápidas e assertivas;
 - Reagir mais rapidamente a mudanças do ambiente;
 - Obter vantagem competitiva;



Fundamentos de Data Warehouse

Contexto:

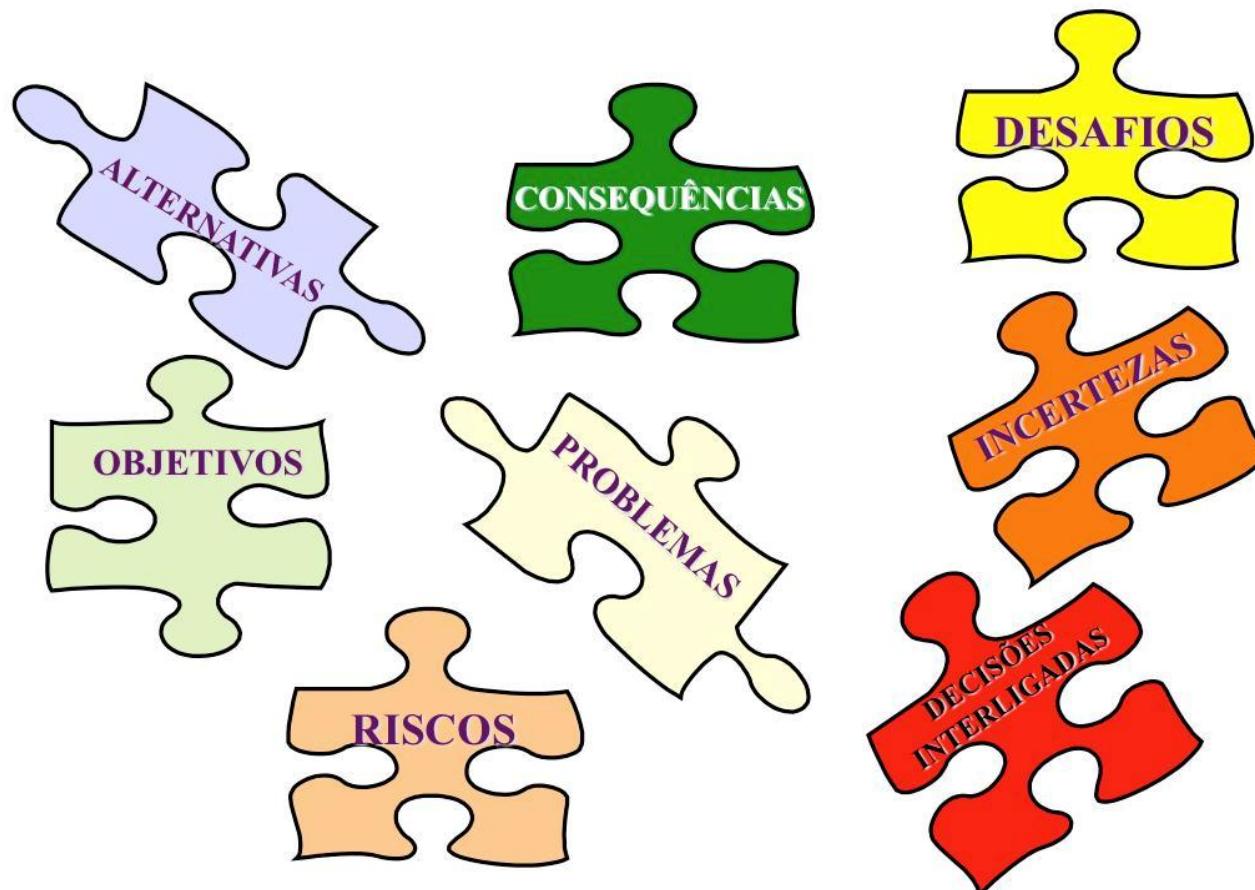
- A necessidade de cruzar informações para realizar uma gestão empresarial eficiente é uma realidade tão verdadeira quanto no passado o foi descobrir se a alta da maré iria propiciar uma pescaria mais abundante.



Fundamentos de Data Warehouse

Contexto:

- No entanto existem algumas variáveis a serem consideradas:



Fundamentos de Data Warehouse

Contexto:

- É preciso também saber que:

DECISÃO



**A tomada de decisão
ocorre sempre em
reação a um problema
ou oportunidade.**

Fundamentos de Data Warehouse

Contexto:



Fundamentos de Data Warehouse

Contexto:

- Em outro extremo temos as informações, porém:
 - Disponíveis de maneira não integrada;
 - ❖(sistemas/arquivos não integrados).
 - Segmentada em diferentes ambientes, com diferentes focos;
 - ❖(espalhados em diferentes e múltiplas plataformas)
 - Dificuldade de análise;
 - ❖Utilização de ferramenta inadequada;



Finance



Sales



Engineering



Manufacturing

Fundamentos de Data Warehouse

O que você faria?



Fundamentos de Data Warehouse

Histórico do Data Warehousing

- Década de 80:
 - Organizações perceberam a importância da utilização dos dados para outros fins além do operacional. Era preciso gerar inteligência a partir deles;
 - A inteligência justificaria as decisões passadas, mas também apoiaria na tomada de decisões futuras;
 - O termo *Business Intelligence* tornou-se popular e no final da década de 80 os pesquisadores da IBM Barry Delvin e Paul Murphy desenvolveram o conceito de *Business Data Warehouse*;

Fundamentos de Data Warehouse

Histórico do Data Warehousing

- Década de 80:
 - As primeiras aplicações BI surgiram, porém percebeu-se a necessidade de que os dados operacionais precisavam passar por um processo de transformação, para daí poderem ser utilizados para derivar inteligência.



Fundamentos de Data Warehouse

Histórico do Data Warehousing

- Década de 80:
 - Este novo banco de dados seria:
 - Bem maior que os bancos operacionais;
 - Teria um projeto diferenciado de construção, ideal para manipular grandes quantidades de dados;
 - Armazenaria o passado e o presente da organização.



Fundamentos de Data Warehouse

Conceitos Iniciais:

Mas o que é um Data Warehouse?

Um warehouse (armazém, em português), em geral, é um enorme repositório de mercadorias, essencialmente para o armazenamento. No contexto de um Data Warehouse, como o nome sugere, esta mercadoria é o dado.

Assim, um Data Warehouse armazena os dados de uma fonte central, normalmente um banco de dados transacional, e distribui esses dados de uma forma que permite análises fáceis e geração de relatórios.

**E qual a diferença com o
BD Tradicional?**



Fundamentos de Data Warehouse

Conceitos Iniciais:

Para responder precisamos relembrar qual o seu propósito do DW!

- **Objetivos do DW:**
 - Satisfazer necessidades de análise de informações;
 - Monitorar e comparar situações atuais com passadas;
 - Estimar situações futuras.
- **Diferenças essenciais:**
 - Volume de dados diferente;
 - Forma de armazenamento diferente.



Fundamentos de Data Warehouse

Conceitos Iniciais:

Data Warehouse: “Uma coleção de dados, orientados ao assunto, integrados, não voláteis e variantes no tempo, que fornece suporte ao processo de tomada de decisões na organização”.

(INMON, 1992)

- Orientados a assunto;
- Integrados;
- Não voláteis;
- Variantes no tempo;

Fundamentos de Data Warehouse

Conceitos Iniciais:

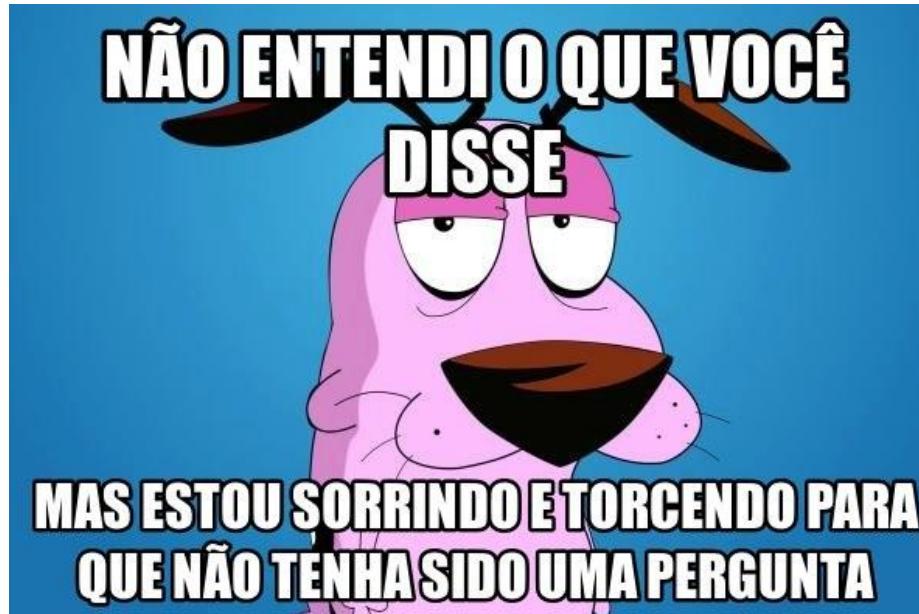
- **Orientados a assuntos:** O DW está orientado ao redor do principal assunto da organização, por exemplo, vendas de produtos a diferentes tipos de clientes, atendimentos e diagnósticos de pacientes, rendimento de estudantes;
- **Integrados:** diferentes nomenclaturas, formatos e estruturas das fontes de dados precisam ser acomodadas em um único esquema para prover uma visão unificada e consistente da informação;
- **Não voláteis:** os dados de uma data warehouse não são modificados como em sistemas transacionais (exceto para correções), mas somente carregados e acessados para leituras, com atualizações apenas periódica;
- **Variantes no Tempo:** o histórico dos dados por um período de tempo superior ao usual em BDs transacionais permite analisar tendências e mudanças.

Fundamentos de Data Warehouse

Conceitos Iniciais:

Mas o que NÃO é um Data Warehouse?

A primeira coisa que devemos aprender é não confundir Data Warehouse com Data Warehousing.



Fundamentos de Data Warehouse

Conceitos Iniciais:

Pense no *Data Warehousing* como uma linha de produção de uma fábrica, na qual o produto final é o *Data Warehouse*.



CUIDADO: Denomina-se de *Data Warehousing* tanto o processo de construção do *Data Warehouse* como também o seu uso.

Fundamentos de Data Warehouse

Conceitos Iniciais:

Mas o que NÃO é um Data Warehousing?

Produto: O *Data Warehousing* não é um produto e não pode ser comprado como um software de banco de dados. O *Data Warehousing* é similar ao desenvolvimento de um ERP, ou seja, ele exige análise do negócio, exige o entendimento do que se quer retirar das informações;

Linguagem: O *Data Warehousing* não pode ser aprendido ou codificado como uma linguagem. Devido ao grande número de componentes e de etapas, um *Data Warehousing* suporta diversas linguagens e programações desde a extração dos dados até a apresentação dos mesmos;

Fundamentos de Data Warehouse

Conceitos Iniciais:

Mas o que NÃO é um *Data Warehousing*?

Projeto: Devido a natureza evolutiva do *Data Warehousing*, é mais fácil aceitá-lo como um processo que está sempre em crescimento do que em um projeto com início-meio-fim, o que definitivamente ele parece mas não é;

Modelagem: O *Data Warehousing* não é apenas modelagem de banco de dados e não é constituído por mais de um modelo. Existe outras etapas que compreende todos os procedimentos de ETL, *Cleansing* e apresentação das informações ao usuário final.



**PUC
GOIÁS**



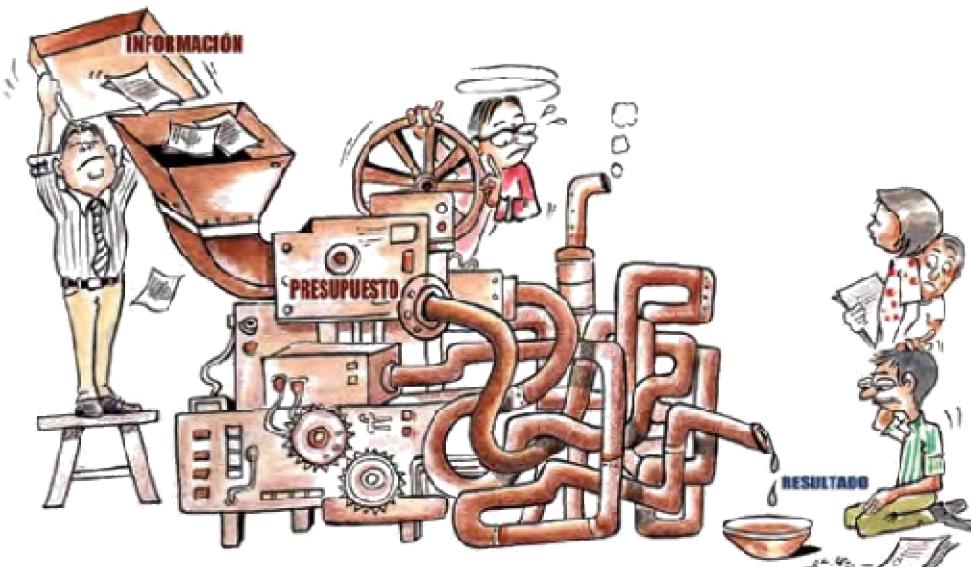
UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Fundamentos de Data Warehouse

Conceitos Iniciais:

Data Mart: “Um subconjunto lógico do Data Warehouse, geralmente visto como um *data warehouse setorial*”.

(KIMBALL, 1994)



Trata-se de um sub-conjunto de dados de um *Data Warehouse* (DW) e geralmente são dados referentes a um assunto em especial (ex: Vendas, Estoque, Controladoria) ou diferentes níveis de sumarização (ex: Vendas Anual, Vendas Mensal, Vendas 5 anos), que focalizam uma ou mais áreas específicas.

Fundamentos de Data Warehouse

Conceitos Iniciais:

Ok. Já entendi o que é Data Warehouse, mas qual a relação com Business Intelligence (BI)?

Antes de explicitarmos a relação entre BI e DW precisamos primeiro definir BI.



Fundamentos de Data Warehouse

Conceitos Iniciais:

Business Intelligence: é uma tecnologia que permite às empresas transformar dados guardados nos seus sistemas em informação qualitativa e importante para a tomada de decisão.

Simplificar



Refere-se a coleta, organização, análise, compartilhamento e monitoramento de informações para suporte a gestão de negócios. Inclui Data Warehouse (DW,) Data Mining (DM), Customer Relationship Management(CRM), dentre outros.

Fundamentos de Data Warehouse

Conceitos Iniciais:

E qual a relação entre BI e DW?

O BI converte o volume de dados, provido pelo DW, em informações relevantes ao negócio, através de relatórios analíticos e outras ferramentas, de maneira a usá-las para a tomada de decisão.



Fundamentos de Data Warehouse

Onde podemos empregar BI?



Fundamentos de Data Warehouse

Benefícios do Uso do BI

- Maior capacidade de análise;
- Maior acompanhamento dos processos de negócio;
- Agilidade na montagem de informações;
- Possibilidade do uso de indicadores de gestão;
- Padronização do acesso aos dados;
- Padronização de regras de negócio;
- Diminuição nos ciclos de fechamento, revisão e liberação de informações;
- Automação de processos de informação.



Fundamentos de Data Warehouse

Contexto das Organizações:

- Lida com “dois mundos de informações diferentes”:
 - Operacional (execução) x Gerencial (Planejamento e Controle);
 - Objetivos distintos;
 - Requisitos de ambientes de BD distintos.



Fundamentos de Data Warehouse

Ambientes de BDs Operacionais:

- Dão suporte às funções associadas à **execução** do negócio da empresa: Sistemas administrativos, de controle de estoque, assinaturas de clientes, etc.
- Processamento: **OLTP** (*On-line transactional processing*);
 - ✓ Transações pontuais (**1 registro por vez**);
 - ✓ Velocidade e automação de funções “*repetitivas*”;
 - ✓ Atualizações e consultas em grande número;
 - ✓ Trabalha com alto nível de detalhe;
 - ✓ Situação corrente;
 - ✓ Altamente Voláteis;
 - ✓ Modelagem E\R.

Fundamentos de Data Warehouse

Ambientes de BDs Gerenciais:

- Dão suporte às atividades de **tomada de decisões gerenciais** na organização: Sistemas de suporte à decisão, ferramentas de análise.
- Tipo de processamento: **OLAP** (*On-line analytical processing*)
 - ✓ Centenas, milhares, ... de registros por consulta;
 - ✓ Diversas fontes de dados;
 - ✓ Diferentes perspectivas;
 - ✓ Operações de agregação e cruzamentos;
 - ✓ Atualização quase inexistente, apenas novas inserções;
 - ✓ Dados históricos são relevantes;
 - ✓ Permitem identificação de perfis, tendências e padrões;

Fundamentos de Data Warehouse

Ambientes de BDs Gerenciais:

- Tipo de processamento: **OLAP** (*On-line analytical processing*)
 - ✓ Redundância de dados aceita;
 - ✓ Alto desempenho na recuperação de dados *versus* economia de espaço;
 - ✓ Modelagem Multidimensional.



Business School Brasil
O futuro é agora e já é amanhã



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Fundamentos de Data Warehouse

BDs Operacionais Versus BDs Gerenciais



Business School Brasil
O futuro é agora. Agora é o seu.



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Fundamentos de Data Warehouse

Talvez ainda não tenha ficado claro porque precisamos de tudo isso.

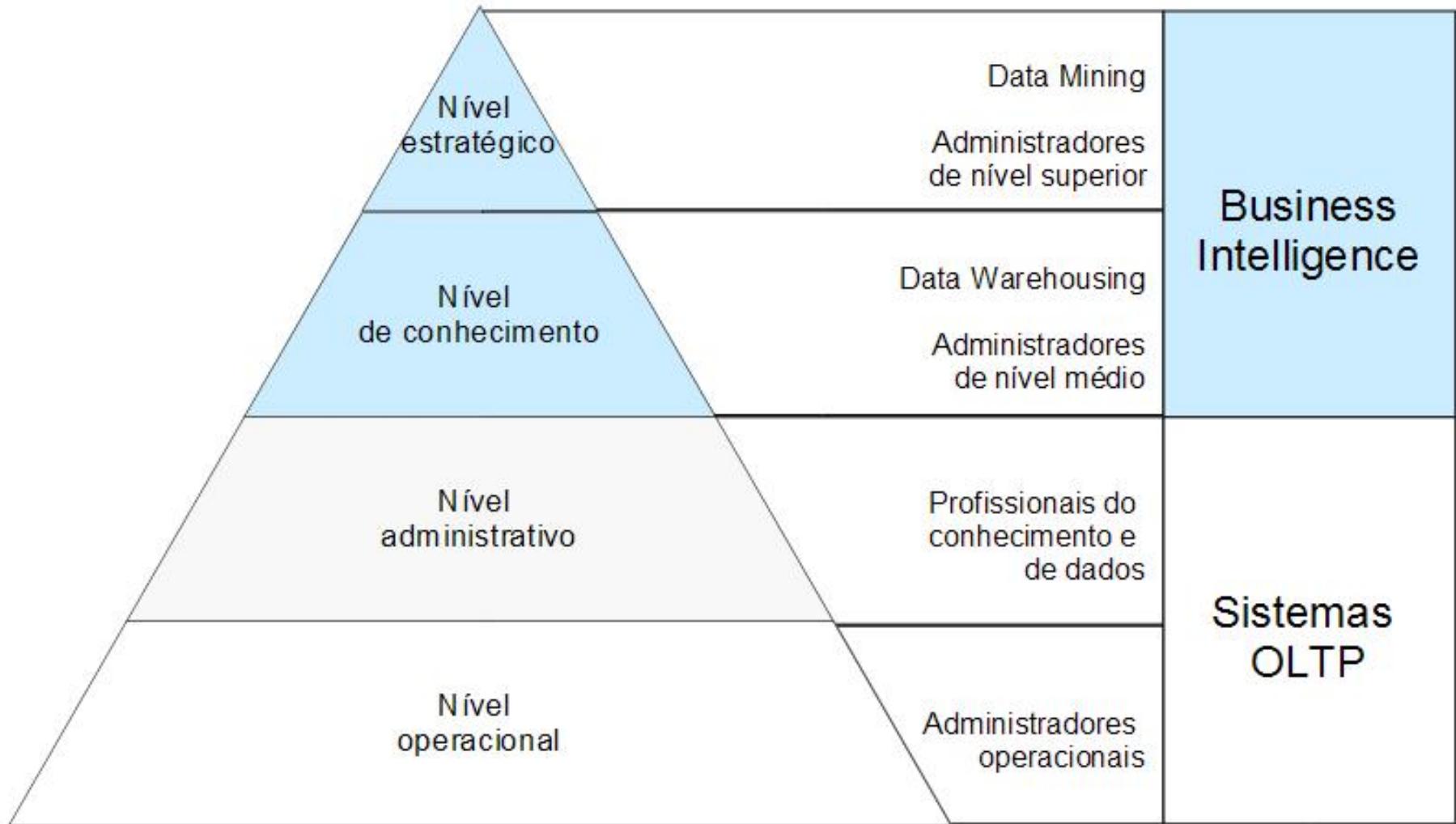
Qual produto tenho que comprar hoje para maximizar o lucro da empresa nas vendas de dezembro?

Em quanto aumentaram as vendas do Produto X durante a promoção em abril de 2003?

Como foi a distribuição das vendas do Produto X em todas as cidades no ano passado?



Fundamentos de Data Warehouse



Conteúdo Programático

UNIDADE II: COMO CONSTRUIR UM DATA WAREHOUSE

- O Ambiente de Data Warehouse;
- Projeto de um Data Warehouse;
- Definição dos Requisitos de Negócio para um DW;
- Granularidade no Data Warehouse;
- Data Warehouse Distribuído.



Business School Brasil
O futuro do negócio é educação



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Como Construir um DW

Projeto e Implementação de DWs

- *Data warehouses* podem requerer muito investimento (tempo, dinheiro, trabalho), dificultando sua adoção em instituições de pequeno e médio porte;
- A modelagem é crítica para o sucesso de um *Data Warehouse* e merece atenção;
- Empreendimentos que não considerem as diferenças entre modelagem de bancos de dados convencionais e data warehouses, incluindo questões técnicas e administrativas, podem facilmente fracassar.

Como Construir um DW

Projeto e Implementação de DWs

Há a necessidade de critérios para planejamento, projeto e implementação de data warehouses!



Como Construir um DW

Critérios para Projeto de DWs

- Determine um escopo pequeno;
- Escolha um departamento;
- Defina com clareza os objetivos;
- Utilize os recursos tecnológicos disponíveis;
- Não proponha um projeto coorporativo;
- Conceba um projeto escalável.



Como Construir um DW

Formas de Desenvolvimento de DWs

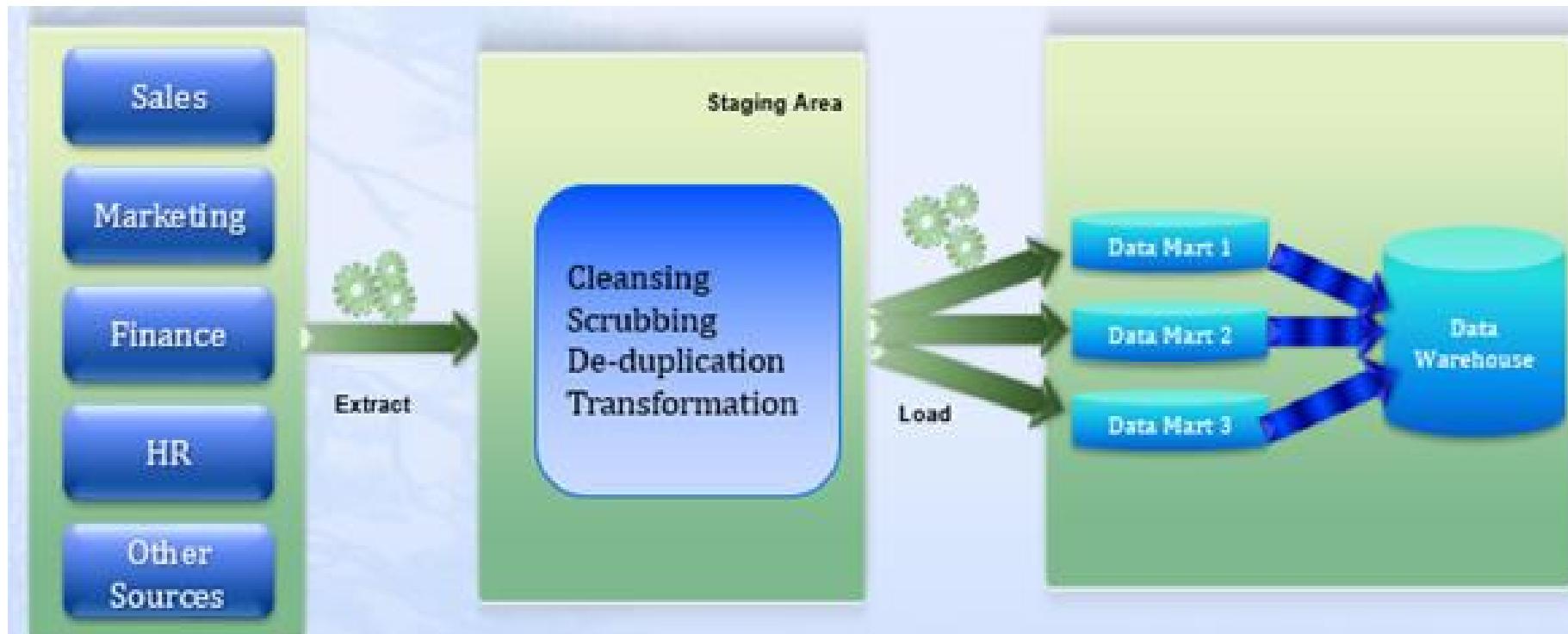
- **Top-down:** Projeto e implementação do DW completo definindo o esquema integrado, fontes de dados e Datamarts;



Como Construir um DW

Formas de Desenvolvimento de DWs

- **Bottom-up:** Projeto e implementação de pequenas DWs ou DMs que vão se integrando aos poucos;



Bottom Up Approach

Como Construir um DW

Localização dos Dados no DW

- **Repositório centralizado (global)**: solução muito utilizada, mas com o inconveniente de requerer investimento em um servidor com alta capacidade de processamento e armazenamento.
- **Repositórios Independentes**: dados armazenados em diferentes locais ou DMs de acordo com áreas de interesse (e.g., financeiro, marketing).
- **Níveis de Detalhes**: dados altamente consolidados/resumidos em um servidor e dados detalhados em outro.

Como Construir um DW

DW Centralizado



Business School Brasil
O futuro é agora, aprendendo



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Como Construir um DW

DW Descentralizado



Como Construir um DW

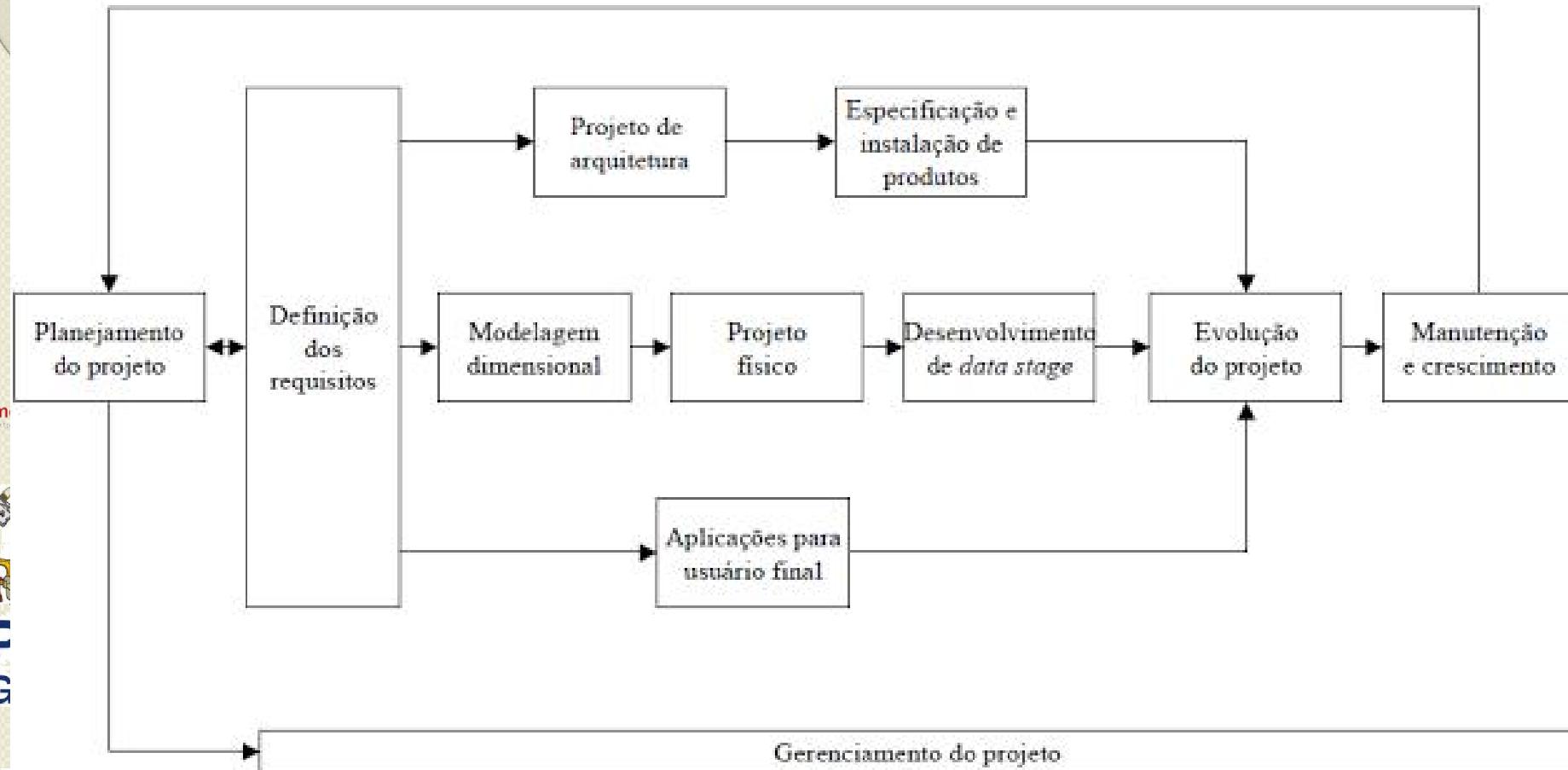
Fases do Desenvolvimento de DWs

- 1. Planejamento**
- 2. Levantamento das necessidades e fontes de dados**
- 3. Integração de dados**
- 4. Modelagem dimensional**
- 5. Projeto físico do banco de dados**
- 6. Projeto das transformações de dados (ETC)**
- 7. Desenvolvimento de aplicações**
- 8. Validação e teste**
- 9. Treinamento**
- 10. Implantação**



Como Construir um DW

Ciclo de Vida de DW

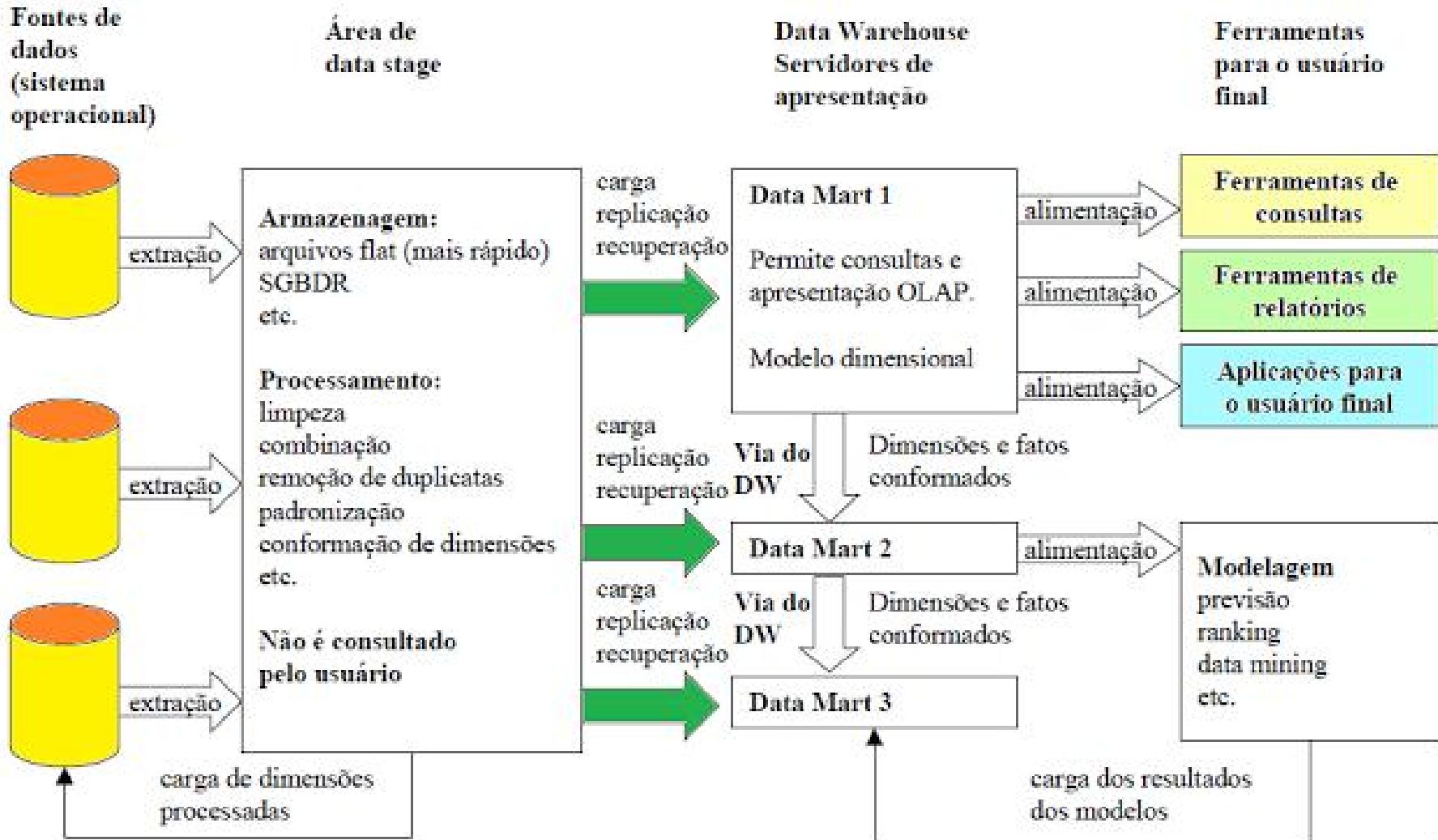


Business Dimensional Lifecycle

Adaptada de Kimball, Ralph; et al. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*, Nova Iorque, John Wiley & Sons, 1998.

Como Construir um DW

Elementos Básicos de um DW



Como Construir um DW

Técnicas que podem ajudar no Planejamento do DW

- Análise de Alto Nível sobre os Requisitos de Negócios.
- Prova de Conceito (POC).



Como Construir um DW

Análise Inicial sobre os Requisitos de Negócio

- Entender as iniciativas estratégicas de negócio;
- Identificar quais indicadores de performance ou métricas para cada iniciativa estratégica;
- Determinar os processos de negócios que devem ser monitorados e os impactos;
- Determinar os benefícios que serão incorporados ao negócio, com um melhor acesso a informação.

Como Construir um DW

Definição de Escopo

- É necessário definir o escopo do projeto sendo esse escopo totalmente justificável.
- Algumas dicas para elaboração do escopo :
 - ❖ Escopo é um trabalho conjunto entre TI e Negócios.
 - ❖ O escopo inicial deve ser significativo e gerenciável.
 - ❖ Focar inicialmente em um processo de negócio de somente uma fonte de dados.
 - ❖ Limite a quantidade de usuários que vai acessar o DW inicialmente.
 - ❖ Defina critérios de sucesso juntamente com o escopo.

Como Construir um DW

Escopo Preliminar - Documentação

- A documentação do escopo é de extrema importância para todo o Ciclo de Vida do DW;
- O entendimento, a aceitação, as exclusões, critérios de sucesso, os riscos e planos de ação, devem sempre estarem declarados e aceitos por todo o grupo de trabalho.



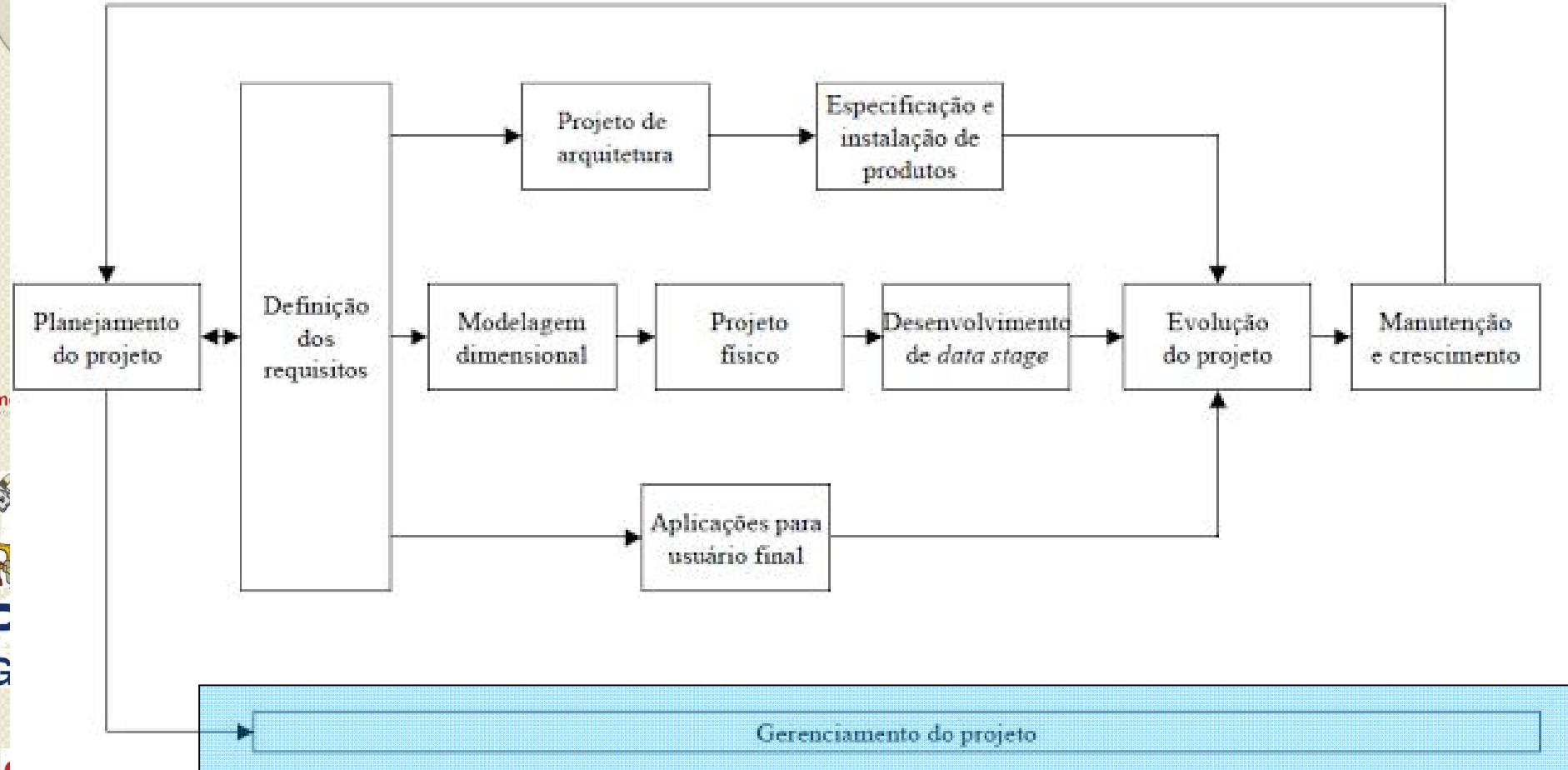
Como Construir um DW

Planejamento do Projeto

- Tarefas :
 - ❖ Determinar a Gerência e a gestão;
 - ❖ Determinar a equipe do projeto (Staff) do Projeto;
 - ❖ Determinar o dia-a-dia do Projeto;
 - ❖ Determinar os Key Users;
- Desenvolver o Plano do Projeto
 - ❖ Cronograma de Atividades e Tarefas.
- Documentos
 - ❖ Documento de Escopo;
 - ❖ Plano de Projeto.

Como Construir um DW

Gerenciamento de Projetos



Business Dimensional Lifecycle

Adaptada de Kimball, Ralph; et al. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*, Nova Iorque, John Wiley & Sons, 1998.

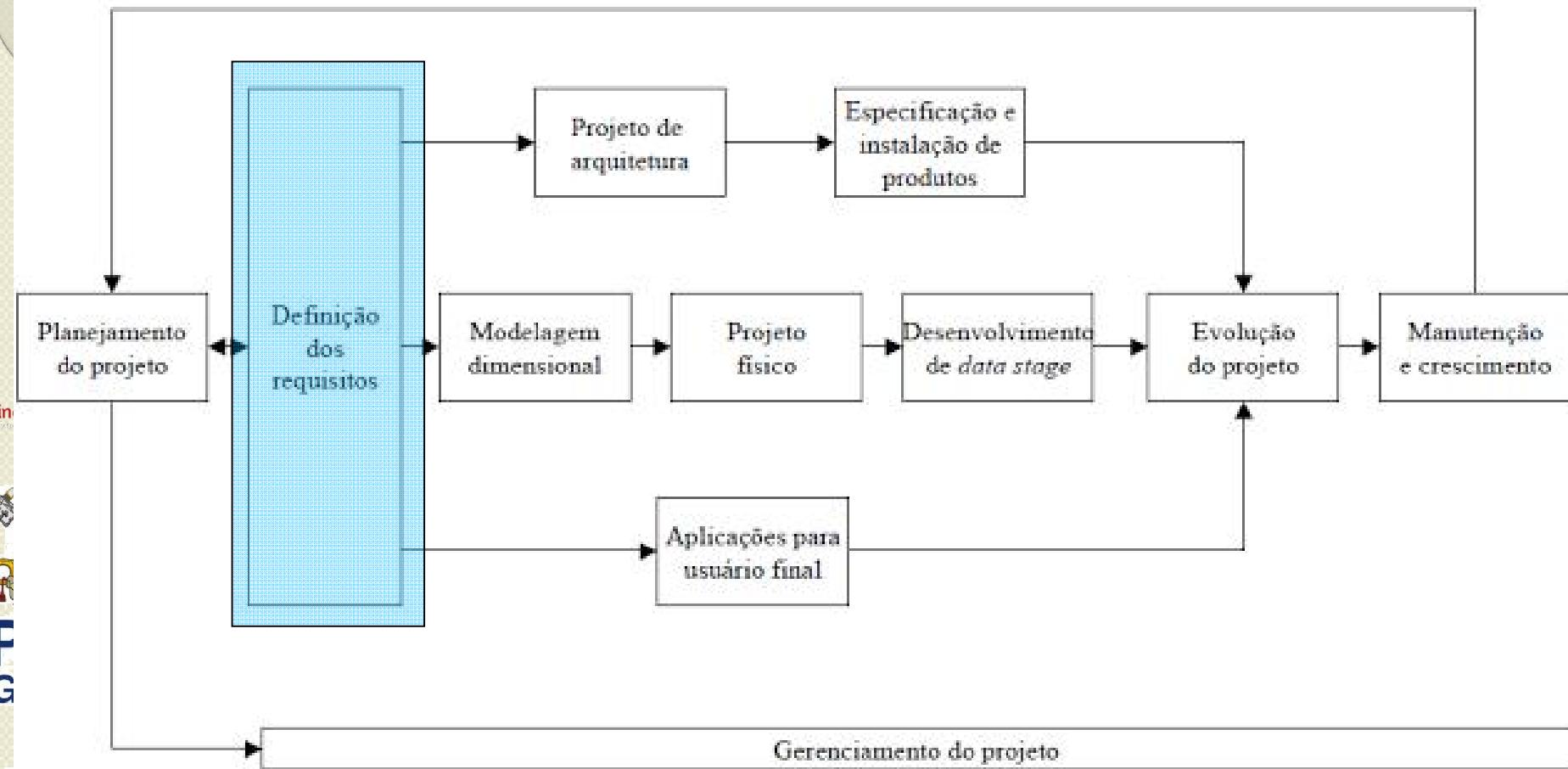
Como Construir um DW

Desafios e Documentos da Fase

- Desafios :
 - ❖ Times multifuncionais, problemas com dados, escopo.
- Documentos :
 - ❖ Relatório de Status;
 - ❖ Ata da Reunião de Status;
 - ❖ Plano de Projeto (Revisão e Atualização);
 - ❖ Planilha de Registro de Problemas e Itens;
 - ❖ Planilha de Controle de Mudanças;
 - ❖ Documento de Requisição de Mudanças.

Como Construir um DW

Definição de Requisitos de Negócio



Business Dimensional Lifecycle

Adaptada de Kimball, Ralph; et al. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*, Nova Iorque, John Wiley & Sons, 1998.

Como Construir um DW

Sugestão para coleta de requisitos:

- Sessão com todos os envolvidos, para:
 - ❖ Definição dos processos de negócios;
 - ❖ Responsáveis;
 - ❖ Visões necessárias;
 - ❖ Sistemas Fontes.
- Entrevistas individuais, para:
 - ❖ Detalhar os processos de negócios;
 - ❖ Sistemas Legados.
- Quem vamos entrevistar:
 - ❖ Gestores de Negócios;
 - ❖ Analistas de Negócios;
 - ❖ Analistas dos Sistemas Legados.

Como Construir um DW

Finalizando o Levantamento de Requisitos:

- Descrição minuciosa das Entrevistas
 - ❖ Transcreva todo o resultado das entrevistas;
 - ❖ Envie para uma revisão dos usuários;
 - ❖ Faça isso de uma maneira clara e intuitiva.
- Documento de Descrição dos Requisitos
 - ❖ Unifique as entrevistas orientadas por assuntos e relate os tópicos mais relevantes com seus entrevistadores;
 - ❖ Monte um documento final de entendimento.

Como Construir um DW

Finalizando o Levantamento de Requisitos:

- **Elabore um documento estruturado com :**
 - ❖ Visão Executiva;
 - ❖ Visão Geral do Projeto;
 - ❖ Requisitos do Negócio:
 - Revisão dos Objetivos do Negócio;
 - Requisitos para análise.
 - ❖ Análise do Sistema Fonte;
 - ❖ Critério de sucesso;

Como Construir um DW

Documentos da Fase:

- **Email de Pré-Entrevista para os usuários;**
- **Questionários:**
 - ❖ Para executivos;
 - ❖ Para Gerentes e Analistas de Negócio;
 - ❖ Para Analistas de Sistemas Legados.
- **Resumo/Transcrição das Entrevistas;**
- **Documento Final de Requisitos.**

Como Construir um DW

Equipe de Desenvolvimento de um DW:



Business School Brasil



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Como Construir um DW

Conceitos Ligados a Qualidade dos Dados

- **Precisão:** grau de correção (e.g., casas decimais);
- **Consistência:** grau de coerência entre os dados;
- **Integridade:** não falta pedaços do banco de dados e não há dados corrompidos;
- **Abrangência:** escopo do banco de dados como um todo (e.g., intervalo de tempo, espaço geográfico coberto, gama de produtos).

Como Construir um DW

Granularidade. O que é isso?

Os DWs estão organizados em diferentes níveis de agregação:

- **nível inferior**: dados primitivos coletados do ambiente operacional;
- **níveis intermediários**: dados com graus de agregação crescente;
- **nível superior**: dados altamente resumidos (agregados).

Devido ao volume de dados armazenados no DW, esses dados podem ser transferidos periodicamente para o nível antigo.

Como Construir um DW

Níveis de Agregação:



Business School Brasil



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Como Construir um DW

Níveis de Agregação:



Business School Brasil

O futuro é agora. Agora é o seu.



**PUC
GOIÁS**



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Como Construir um DW

E agora, o que é *Granularidade*?

Grau de detalhamento em que os dados são armazenados em um nível.

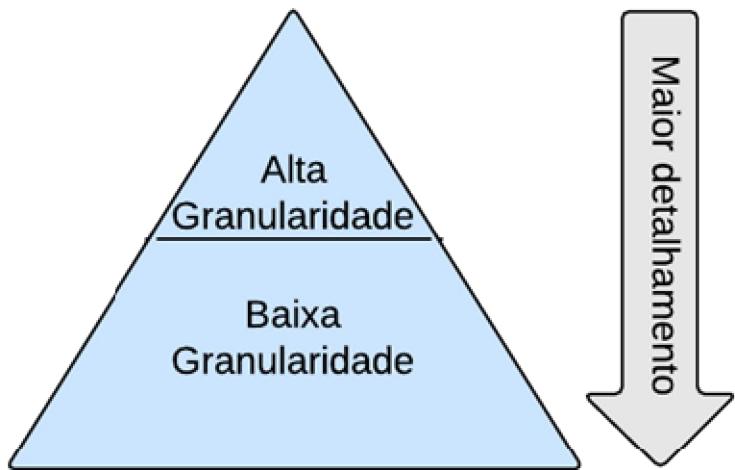
A granularidade é uma questão de projeto muito importante:

- Impacta no volume de dados armazenado;
- Afeta as consultas que podem ser respondidas.



Como Construir um DW

Analisando os efeitos da Granularidade



Grão muito pequeno

- 1.Tamanho do data warehouse é muito grande;
- 2.Praticamente qualquer consulta pode ser respondida.

Grão muito grande

- 1.Tamanho do data warehouse é menor;
- 2.número de consultas que podem ser respondidas é menor.

Conteúdo Programático

CONTEÚDO EXTRA:

- Modelagem de Dados;
- Normalização.



Business School Brasil
O futuro é agora, acredite no seu.



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Modelagem de Dados

O que é Modelagem de dados?

Processo pelo qual você planeja ou projeta a base de dados, de forma que possa construir um banco de dados consistente, de forma a exigir menos espaço em disco e aproveitar os recursos disponíveis no SGBD.



Modelagem de Dados

Por que Modelar os dados?

É importante modelar os dados a fim de conhecer melhor as informações dos usuários e como elas se relacionam, a fim de representar, da melhor forma, o ambiente observado, criando, por consequência, bancos de dados mais corretos e eficientes.



Modelagem de Dados

A modelagem é uma das Etapas do ciclo de desenvolvimento de Sistemas de Bancos de Dados

- **Etapas:**

- ✓ Para poder realizar a modelagem dos dados, você precisa fazer um levantamento de requisitos. Ou seja, precisa investigar quais dados deverão fazer parte do banco de dados, a fim de representar bem o problema a ser resolvido e poder atender as necessidades de armazenamento (persistência) dos dados da aplicação;
- ✓ Uma vez que se saiba os dados que devem ser manipulados, você deve analisar como esses dados podem ser representados e relacionados (olhando para o mundo real) através de um modelo de dados;



Modelagem de Dados

Etapas:

- ✓ Uma vez que os dados estejam modelados o banco de dados será projetado, transformando o modelo de dados criado em estruturas de mais baixo nível que possam ser criadas dentro do SGBD;
- ✓ Posteriormente, o BD é realmente implementado usando algum dos SGBDs disponíveis no mercado e, depois, mantido e monitorado pelo administrador de banco de dados.

Modelagem de Dados

Mas como modelar?



Existem modelos para diferentes níveis de abstração de representação de dados. São eles: modelos conceituais (também conhecido como modelo com base em objetos), modelos lógicos (também conhecido como modelo com base em registros) e modelos físicos.

Modelagem de Dados

- **Modelo Conceitual:**

- ✓ É a primeira etapa da modelagem de dados, sendo a descrição mais abstrata da realidade, modelando de forma mais natural os fatos do mundo real, suas propriedades e relacionamentos;
- ✓ É utilizado para entendimento, transmissão, validação de conceitos, mapeamento do ambiente e para facilitar o diálogo entre usuários e desenvolvedores;
- ✓ A modelagem conceitual do BD independe da sua implementação, não contendo nenhum detalhe da mesma.
- ✓ A modelagem conceitual dos bancos de dados relacionais é feita através da criação do modelo entidade-relacionamento (MER). No caso de bancos de dados orientados a objetos ou objeto-relacional, é usado o modelo de classes da UML (*Unified Modeling Language*).

Modelagem de Dados

- **Modelo Lógico:**

- ✓ Representa os dados em alguma estrutura (lógica) de armazenamento de dados, que vai depender do SGBD a ser utilizado. Ou seja, este modelo vai especificar a representação/declaração dos dados de acordo com o SGBD escolhido;
- ✓ O modelo relacional usa um conjunto de tabelas para representar tanto os dados como a relação entre eles. Cada tabela possui múltiplas colunas e cada coluna possui um nome único.

Modelagem de Dados

- **Modelo Físico:**

- ✓ São usados para descrever os dados no nível mais baixo, tratando de aspectos de implementação do SGBD (como indexação e estruturação de arquivos, controle de concorrência, transações, recuperação em casos de falhas, entre outros);
- ✓ As linguagens e notações para o modelo físico não são padronizadas e variam de produto a produto (são dependentes do SGBD);



Business School Brasil
O futuro é agora e já é amanhã



PUC
GOIÁS



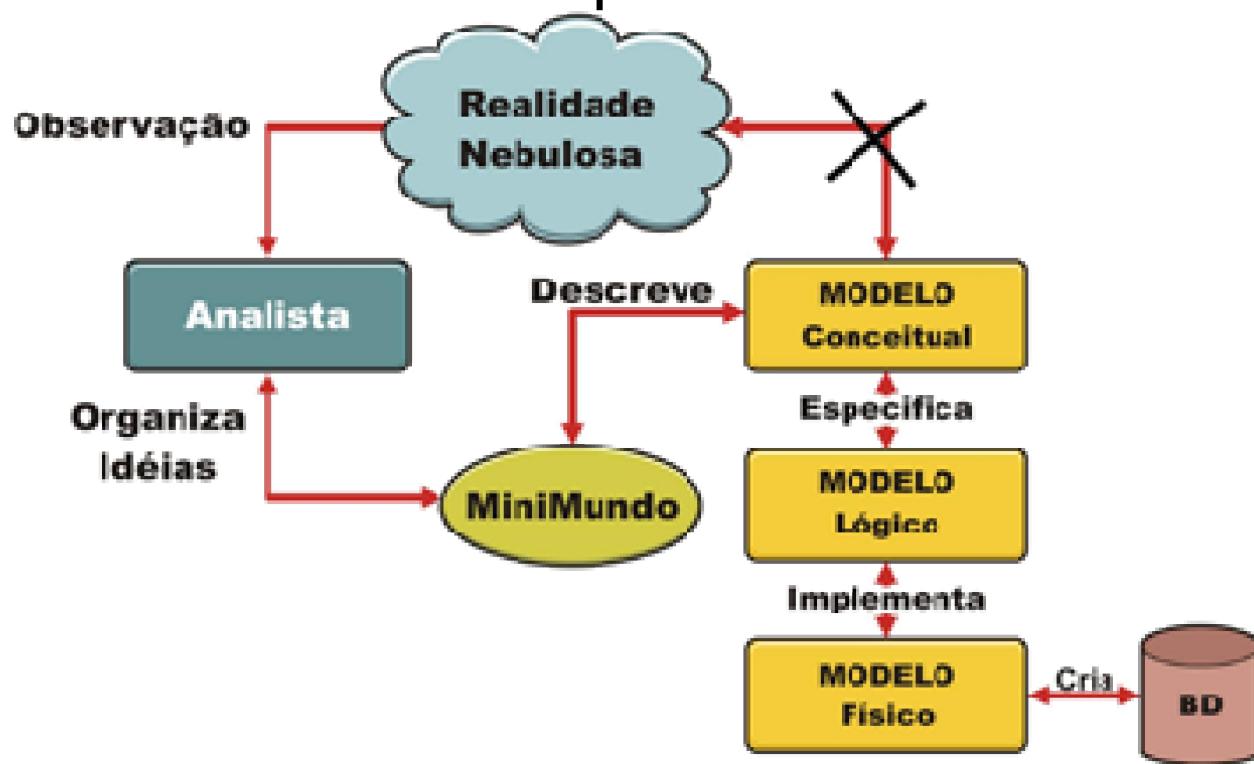
UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Modelagem de Dados

- Todos esses modelos, na verdade, são visões diferentes, com nível de profundidade diferente para os mesmos dados.
 - É importante saber que, a partir de um modelo, o modelo seguinte pode ser derivado.
- ✓ Analista de banco de dados, observa a realidade e, a partir do problema a ser resolvido (aplicação a ser desenvolvida), ele organiza suas idéias e cria um minimundo (que é um subconjunto da realidade contendo os dados necessários para a resolução do problema sendo tratado).

Modelagem de Dados

- ✓ O minimundo tem dados que vão ajudar a descrever o modelo conceitual (mais alto nível de abstração), o modelo lógico é especificado a partir do modelo conceitual e é implementado pelo modelo físico (que é quem realmente é usado para criar o banco de dados (BD)).



Modelagem de Dados

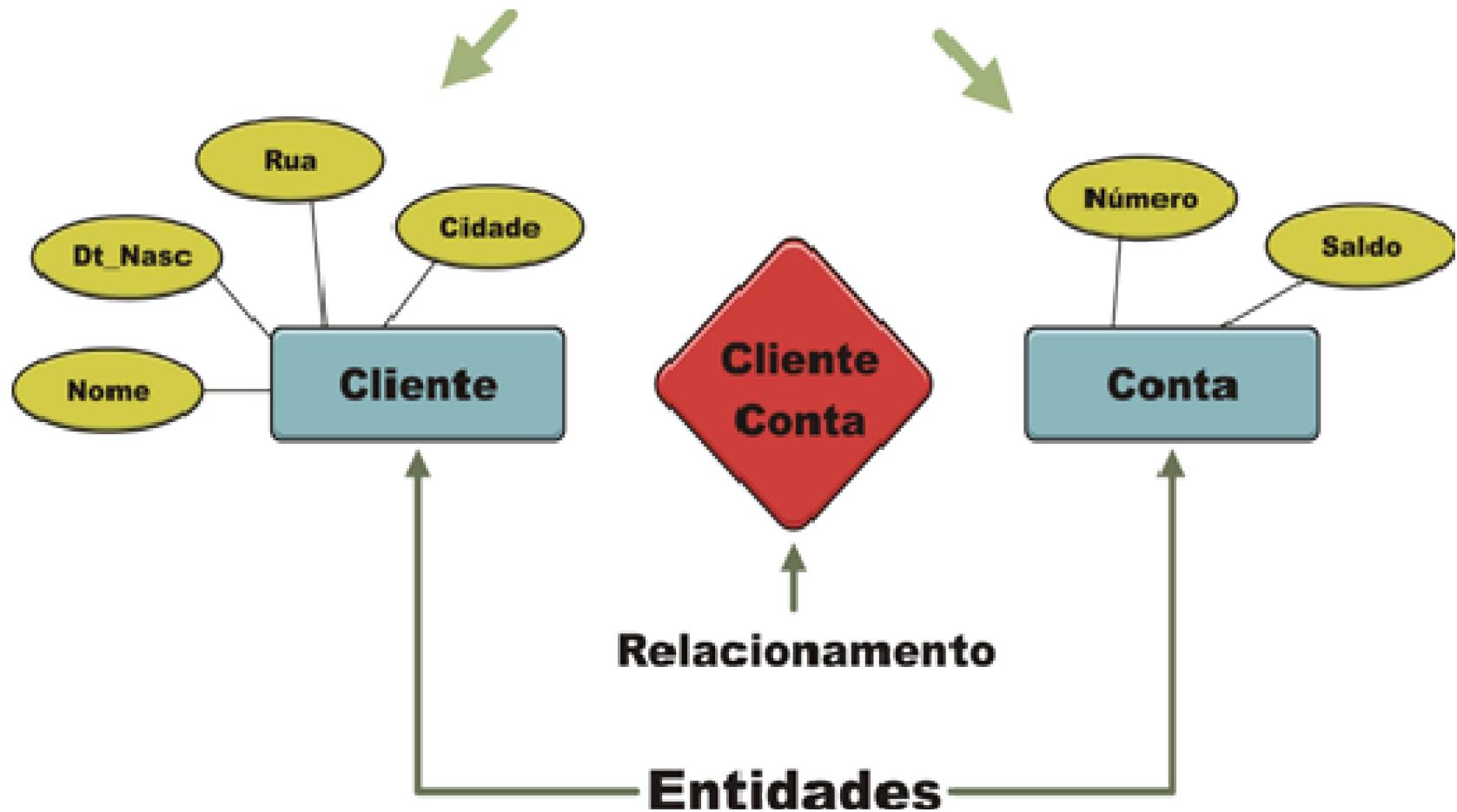
- **Modelo Entidade-Relacionamento:**

- ✓ O modelo entidade-relacionamento (conhecido como MER) foi originalmente definido por Peter Chen em 1976 e é baseado na teoria relacional criada em 1970 por Codd;
- ✓ Atualmente, ele é o modelo de dados conceitual mais difundido e utilizado para modelagem de bancos de dados relacionais;
- ✓ Para iniciar o projeto conceitual do BD, deve ter sido antes definido qual o problema a ser resolvido, ou seja, deve-se ter determinado as fronteiras que delimitam e restringem o minimundo a ser modelado;
- ✓ o MER tem por base a percepção de que o mundo real é formado por um conjunto de objetos chamados entidades, pelo conjunto de atributos e pelo conjunto dos relacionamentos entre esses objetos.



Modelagem de Dados

- Modelo Entidade-Relacionamento:
Atributos



Modelagem de Dados

- **Desenhando o MER:**

- ❖ **Verificação do Modelo Criado:**

- ✓ Um bom modelo ER deve:

- Ser completo;
 - Ser correto;
 - Ser livre de redundância;
 - Refletir o Aspecto Temporal, quando necessário;
 - Evitar entidades isoladas.

Modelagem Lógica



- No projeto de Banco de Dados, a modelagem lógica ou projeto lógico é a terceira etapa;
- O produto dessa etapa é o modelo relacional ou esquema relacional;
- Esse modelo já é dependente do SGBD que for ser escolhido para a implementação do banco de dados.

Modelagem Lógica

Mas, o que vem a ser o modelo lógico?

- É um modelo que vai especificar a representação/declaração dos dados de acordo com o SGBD escolhido, definindo assim a estrutura de registros do BD (onde cada registro define número fixo de campos (atributos) e cada campo possui tamanho fixo);
- Um exemplo de modelo lógico é o modelo relacional (MR);
- Os SGBDs que utilizam o MR são denominados SGBDs Relacionais.

Modelagem Lógica

- **Objetivos do Modelo Relacional:**

- ❖ Prover esquemas de fácil utilização;
- ❖ Melhorar a independência lógica e física de dados;
- ❖ Prover os usuários com linguagens de manipulação de BD de alto nível, permitindo o seu uso por usuários não experientes;
- ❖ Otimizar o acesso aos BDs;
- ❖ Melhorar a integridade e segurança dos dados.

Modelagem Lógica

- **Conceitos do Modelo Relacional:**
 - ❖ Na terminologia do modelo relacional, cada tabela é chamada **relação** e vai possuir um nome único que a identifica;
 - ❖ Cada linha da tabela é chamada **tupla**;
 - ❖ Cada cabeçalho de coluna é conhecido como **atributo**;

Nome da Relação		Atributos				
Aluno	Tuplas	Nome	CPF	Telefone	Endereço	Idade
	Alexandre Silva	987890765-78	3245-9879	R. Jorge Aragão, 2345	23	
	Barbara Alves	435768243-65	3567-5674	R. da Alvorada, 212	30	
	Joana Alves	231435221-90	3456-1232	Av. Alvaro Pires, 34	20	
	Marcos Pontes	675473948-46	3422-7659	R. das Flores, 233	22	
	Regina Marques	567890654-56	3145-8432	Av. Pedro I, 890	18	

Modelagem Lógica

- **Regras de Integridade:**
 - ✓ O modelo relacional, ao definir conceitos como Tabela, Tupla, Atributo, Nulo, Domínio, Chave Primária e Chave Estrangeira deixa implícitas algumas regras fundamentais para a manutenção da consistência do banco de dado;
 - ✓ Essas Regras de Integridade e tratam dos cuidados que analistas, projetistas e programadores devem observar ao implementar as rotinas de Inclusão, Alteração e Exclusão de dados nas bases de dados;
 - ✓ Na prática, as restrições de integridade fornecem meios para assegurar que mudanças feitas no banco de dados não resultem na perda da consistência sobre estes dados.
 - ✓ Em DW estas restrições são desligadas no processo ETL!



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Normalização

- **Introdução:**
 - ❖ **Projetar um banco de dados relacional significa agrupar atributos para formar bons esquemas de relações**

Mas o que seria um bom esquema?

- ❖ **Em nível geral, poderíamos dizer que seria um esquema fácil de entender e em que as tuplas da relação fossem armazenadas e acessadas de forma eficiente.**
 - ✓ É preciso que sejam minimizadas, ao máximo, a redundância nos dados e as anomalias de inserção, atualização e exclusão;
 - ✓ É preciso garantir a integridade dos dados, evitando que informações sem sentido sejam inseridas;
 - ✓ É preciso organizar e dividir as tabelas da forma mais eficiente possível.

Normalização

- **Dependências Funcionais:**
 - ❖ Sempre que um atributo **X** identifica um atributo **Y** dizemos que entre eles há uma dependência funcional, ou seja, se o valor de um atributo **X** permite descobrir o valor de um outro atributo **Y**, dizemos que **X** determina funcionalmente **Y**;
 - ❖ Portanto, que **X** é o determinante e que **Y** é o dependente;
 - ✓ A representação é: $X \rightarrow Y$;
 - ❖ Por exemplo, dada uma determinada cidade (não considerando cidades homônimas) sabemos o seu estado e com o estado temos o país:
 - ✓ Cidade \rightarrow estado;
 - ✓ Estado \rightarrow país;

Normalização

• Dependências Funcionais:

- ❖ A dependência funcional é caracterizada pela existência de campos em uma determinada tabela relacional cuja ocorrência de valores está associada a valores que são preenchidos em outros campos na mesma tabela;
- ❖ Por exemplo, suponha uma tabela **EMPREGADO** que possui dois atributos **CPF** e **NOME**.
- ❖ O atributo **NOME** é funcionalmente dependente do atributo **CPF**. Assim, **CPF → NOME**;
- ❖ Desta forma, o nome é função do **CPF**, ou seja, se eu tiver um número de **CPF**, poderei encontrar o nome da pessoa correspondente;

Normalização

- **Dependências Funcionais:**
 - ❖ **Dependência Parcial:**
 - ✓ Ocorre quando a chave primária é composta e existe um campo da relação que depende somente de parte desta chave primária composta;

<u>CPF_Empregado</u> (PK)	<u>Cod_Projeto</u> (PK)	Nome_Empregado	Nome_Projeto	Horas_Trabalhadas
123456	11	Ana Maria Gomes	SoftHouse	40
654321	22	José da Silva	HardCore	20
789654	33	Cláudio Alencar	LinuxP	40

- ✓ O campo Nome_Empregado depende apenas de parte da chave primária (no caso, do CPF_Empregado);

Normalização

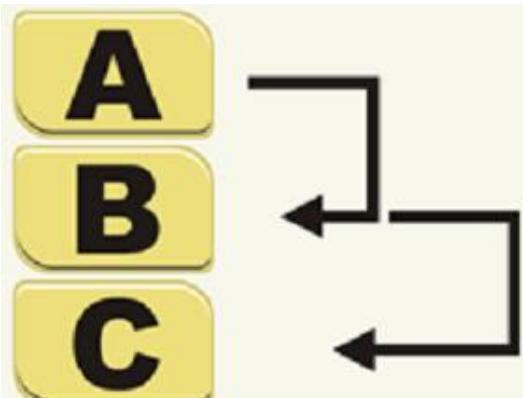
- **Dependências Funcionais:**
 - ❖ **Dependência Parcial:**

<u>CPF_Empregado</u> (PK)	<u>Cod_Projeto</u> (PK)	Nome_Empregado	Nome_Projeto	Horas_Trabalhadas
123456	11	Ana Maria Gomes	SoftHouse	40
654321	22	José da Silva	HardCore	20
789654	33	Cláudio Alencar	LinuxP	40

- ✓ O mesmo ocorre com o atributo Nome_Projeto que só depende do atributo Cod_Projeto;
- ✓ Apenas o atributo Horas_Trabalhadas é funcionalmente dependente da chave primária completa (porque para determinar as horas trabalhadas, precisamos saber o CPF do empregado e para qual projeto ele está trabalhando através do código do projeto).

Normalização

- **Dependências Funcionais:**
 - ❖ **Dependência Transitiva:**
 - ✓ Ocorre quando uma coluna depende não somente da chave primária da tabela, mas também de uma segunda coluna (ou conjunto de colunas) da tabela, que não fazem parte da chave primária;
 - ✓ Assim, se um atributo C depende funcionalmente de um atributo B e o atributo B depende funcionalmente de um atributo A, então diz-se que o atributo C depende indiretamente (transitivamente) do atributo A;



Normalização

- **Dependências Funcionais:**
 - ❖ Dependência Transitiva:

<u>Cod_Pedido</u> (PK)	Data_Pedido	Situação_Pedido	CPF_Cliente	Nome_Cliente
1	18/03/2010	Pendente	123456	Pedro Alves
2	22/02/2010	Entregue	654211	Carolina Dantas
3	10/01/2010	Entregue	987654	Olívia Duncan

- ✓ Por exemplo, a relação Pedido_Venda tem como chave primária o atributo Cod_Pedido. Os atributos Data_Pedido, Situação_Pedido e CPF_Cliente dependem funcionalmente dessa chave primária.
- ✓ Porém, o atributo Nome_Cliente depende funcionalmente do CPF_Cliente (que não é a chave primária) e, apenas, indiretamente do Cod_Pedido, o que é uma anomalia, visto que, todos os atributos de uma relação deveriam depender funcionalmente apenas da sua chave primária;

Normalização

- **Normalização (Conceito):**
 - ❖ A normalização de dados pode ser vista como o processo de análise de determinados esquemas de relações com base em suas dependências funcionais e chaves primárias para alcançar as propriedades desejáveis de:
 - Minimização de redundâncias;
 - Minimização de anomalias de inserção, exclusão e alteração;
 - ❖ Nesse processo, os esquemas de relações insatisfatórios, são decompostos em esquemas de relações menores até possuirem as propriedades desejadas:
 - ✓ Quando uma tabela não atende ao critério de uma forma normal, sua estrutura é redesenhada através da projeção (jogar para fora) de alguns atributos, levando a construção de novas tabelas contendo algum atributo que possa refazer o conteúdo da tabela original através da junção (reunir) das tabelas resultantes.

Normalização

- **Normalização:**
 - ❖ **Funções do processo :**
 - ✓ Analisar tabelas e organizá-las de forma que a sua estrutura seja simples, relacional e estável, para que o gerenciamento delas possa ser também simples, eficiente e seguro;
 - ✓ Evitar a perda e a repetição da informação;
 - ✓ Conseguir uma forma de representação adequada para o que se deseja armazenar;
 - ✓ Oferecer mecanismos para analisar o projeto do BD (identificação de erros e possibilidades de melhorias) e oferecer métodos para corrigir problemas que, por ventura, sejam encontrados.

Normalização



Normalização:

❖ Objetivos do processo:

- ✓ Garantir a integridade dos dados, evitando que informações sem sentido sejam inseridas no banco de dados;
- ✓ Organizar e dividir as tabelas da forma mais eficiente possível, diminuindo a redundância e permitindo a evolução do banco de dados com o mínimo de efeito colateral.

❖ Inicialmente Codd propôs três formas normais que ele chamou de primeira (1FN), segunda (2FN) e terceira (3FN) forma normal.

❖ Uma definição mais forte da 3FN (chamada forma normal BOYCE-CODD - FNBC ou BCNF) foi depois proposta por Boyce e Codd.

Normalização

- **Normalização (I FN):**
 - ❖ Para o Modelo Relacional a Forma Normal mais importante consiste da Primeira Forma Normal, pois é considerada parte da própria definição de uma relação;
 - ❖ Uma relação se encontra na Primeira Forma Normal (1FN) se todos os domínios de atributos possuem apenas valores atômicos (indivisíveis), e que o valor de cada atributo na tupla seja um valor simples;
 - ❖ a 1FN não permite a construção de relações que apresentem atributos compostos e nem possibilita a existência de atributos multivalorados em suas tuplas;
 - ❖ Os únicos valores de atributos permitidos devem ser simples e atômicos;

Normalização

Normalização (1 FN):

Relação Paciente

<u>CPF</u> (PK)	Nome	Tipo_Sanguineo	Graus_Lentes
123456	Jonas Borges	AB+	0.25, 0.50
987659	Cássia Lima	O-	1.0, 0.5
007543	Túlio Gomes	A+	0.0, 0.25

**Não
estão na
IFN**

Relação Matricula_Aluno

<u>CPF</u> (PK)	Nome	Cursos
123456	Jonas Borges	Programador, Arquiteto
987659	Cássia Lima	Analista
007543	Túlio Gomes	Operador, Programador, Analista

Normalização

- **Normalização (I FN):**
 - ❖ **Para normalizar para a Primeira Forma Normal deve-se:**
 - **Atributos Compostos** - cada um dos atributos compostos (ou não atômicos) devem ser “divididos” em seus atributos componentes. Por exemplo, o atributo “Graus_Lentes” poderia ser subdivido em Grau_LenteEsq e Grau_LenteDir, quebrando assim o atributo composto nos dois outros que o compõe;

<u>CPF (PK)</u>	Nome	Tipo_Sanguineo	Grau_LenteEsq	Grau_LenteDir
123456	Jonas Borges	AB+	0.25	0.50
987659	Cássia Lima	O-	1.0	0.5
007543	Túlio Gomes	A+	0.0	0.25

Normalização

Normalização (I FN):

- ❖ **Para normalizar para a Primeira Forma Normal deve-se:**

I. Atributos Multi-valorados – temos dois tratamentos possíveis:

- Quando a quantidade de valores a serem preenchidos no atributo multivalorado for pequena e conhecida a priori, substitui-se o atributo multivalorado por um conjunto de atributos de mesmo domínio, cada um monovalorado; Por exemplo, na relação Matricula_Aluno, suponha que a pessoa possa se matricular, no máximo, em três cursos. Dessa forma, poderíamos decompor o atributo Cursos em três colunas.

<u>CPF (PK)</u>	Nome	Curso1	Curso2	Curso3
123456	Jonas Borges	Programador	Arquiteto	Null
987659	Cássia Lima	Analista	Null	Null
007543	Túlio Gomes	Operador	Programador	Analista

Normalização

- **Normalização (1 FN):**
 - ❖ **Para normalizar para a Primeira Forma Normal deve-se:**
 - **Atributos Multi-valorados** – temos dois tratamentos possíveis:
 - Quando a quantidade de valores for muito variável, desconhecida ou grande, retira-se da relação o atributo multivalorado, e cria-se uma nova relação que tem o mesmo conjunto de atributos chave, mais o atributo multivalorado. Por exemplo, na relação Matricula_Aluno, se não soubéssemos a quantidade de cursos nos quais um aluno pode se matricular, deveríamos tirar o atributo Cursos da relação Matricula_Aluno. Depois, criar uma nova relação, que chamamos Aluno_Curso, onde seria colocada a chave primária da relação Matricula_Aluno (no caso, o atributo CPF) e o atributo multivalorado (agora, como monovalorado), também como chave primária (no caso, o atributo Curso);

Normalização

- **Normalização (I FN):**
 - ❖ **Para normalizar para a Primeira Forma Normal deve-se:**

Relação Matricula_Aluno (fora da 1FN)

CPF (PK)	Nome	Cursos
123456	Jonas Borges	Programador, Arquiteto
987659	Cássia Lima	Analista
007543	Túlio Gomes	Operador, Programador, Analista

Relação Aluno_Curso

CPF (PK)	Curso (PK)
123456	Programador
123456	Arquiteto
987659	Analista
007543	Operador
007543	Programador
007543	Analista

Relação Matricula_Aluno

CPF (PK)	Nome
123456	Jonas Borges
987659	Cássia Lima
007543	Túlio Gomes

Normalização

- **Normalização (2 FN):**
 - ❖ Uma relação está na segunda forma normal quando duas condições são satisfeitas:
 - ✓ A relação está na primeira forma normal;
 - ✓ Todos os atributos que não fazem parte da chave-primária dependem funcionalmente de toda a chave primária, ou seja, nenhum dos atributos da relação possui dependência parcial. Em outras palavras, todo atributo A da relação R não é parcialmente dependente da chave-primária de R.
 - ❖ Relações na 1FN e que possuam chave primária simples estão, automaticamente, na 2FN;

Normalização

- **Normalização (2 FN):**
 - ❖ Vejamos agora a seguinte relação:

<u>CPF_Empregado</u> (PK)	<u>Cod_Projeto</u> (PK)	<u>Nome_Empregado</u>	<u>Nome_Projeto</u>	<u>Horas_Trabalhadas</u>
123456	11	Ana Maria Gomes	SoftHouse	40
654321	22	José da Silva	HardCore	20
789654	33	Cláudio Alencar	LinuxP	40

- ❖ A relação Alocação está na 1FN porque não possui atributos multivalorados ou compostos;
- ❖ Porém, não está na 2FN porque o campo Nome_Empregado depende apenas de parte da chave primária (no caso, do CPF_Empregado) e o atributo Nome_Projeto só depende do atributo Cod_Projeto;

Normalização

Normalização (2 FN):

- ❖ **Para normalizar para a Segunda Forma Normal deve-se:**
 - ✓ Para passarmos uma entidade da 1FN para a 2FN, devemos eliminar as dependências parciais:
 - a) Para cada atributo que não faça parte da chave primária da relação, analisar se existe dependência parcial da chave primária. Para identificar a dependência parcial de uma coluna em relação à chave primária, deve-se indagar: Para que o valor do atributo seja determinado, quais as partes da chave primária que devem ser conhecidas?
 - b) Para cada grupo de atributos que tenham a mesma dependência parcial da chave primária, devem-se criar novas relações (tabelas) que terão como chave primária a chave parcial que estava na relação original e todo o grupo de atributos que depende da mesma chave parcial;

Normalização

Normalização (2 FN):

- ❖ **Para normalizar para a Segunda Forma Normal deve-se:**
 - ✓ Para passarmos uma entidade da 1FN para a 2FN, devemos eliminar as dependências parciais:
 - c) Excluir da relação original o grupo de atributos para o qual foi criada uma nova relação;
 - d) Repetir esses procedimentos para cada grupo de atributos que tenha dependência parcial da chave primária da relação original, até que todas as relações somente contenham atributos que dependam de toda a chave primária.

Normalização

Normalização (2 FN):

- ❖ **Para normalizar para a Segunda Forma Normal deve-se:**
- ✓ Por exemplo, a relação Alocação daria origem a duas novas relações, uma vez que há dois grupos e dependências parciais nesta relação: Cod_Projeto e NomeProjeto e CPF_Empregado e Nome_Empregado;

<u>CPF_Empregado</u> (PK)	<u>Cod_Projeto</u> (PK)	<u>Nome_Empregado</u>	<u>Nome_Projeto</u>	<u>Horas_Trabalhadas</u>
123456	11	Ana Maria Gomes	SoftHouse	40
654321	22	José da Silva	HardCore	20
789654	33	Cláudio Alencar	LinuxP	40

Normalização

- **Normalização (2 FN):**
 - ❖ Para normalizar para a Segunda Forma Normal deve-se:

<u>CPF_Empregado</u> (PK)	<u>Cod_Projeto</u> (PK)	<u>Nome_Empregado</u>	<u>Nome_Projeto</u>	<u>Horas_Trabalhadas</u>
123456	11	Ana Maria Gomes	SoftHouse	40
654321	22	José da Silva	HardCore	20
789654	33	Cláudio Alencar	LinuxP	40

<u>CPF_Empregado</u> (PK)	<u>Nome_Empregado</u>
123456	Ana Maria Gomes
654321	José da Silva
789654	Cláudio Alencar

<u>Cod_Projeto</u> (PK)	<u>Nome_Projeto</u>
11	SoftHouse
22	HardCore
33	LinuxP

<u>CPF_Empregado</u> (PK)	<u>Cod_Projeto</u> (PK)	<u>Horas_Trabalhadas</u>
123456	11	40
654321	22	20
789654	33	40

Normalização

- **Normalização (2 FN):**
 - ❖ **Para normalizar para a Segunda Forma Normal deve-se:**
 - ✓ Relações que não estão na 2FN podem apresentar problemas de inconsistência devido à duplicidade dos dados e à perda de dados em operações de remoção/alteração;
 - ❖ **Quais são os problemas da relação abaixo?**

Cod_Turma (PK)	Sigla_Disciplina (PK)	Num_SalaAula	Num_Horas
11	IP	S15	6
22	IP	S16	6
11	BD	S02	4
22	SO	S10	4
22	SO	S12	4

Normalização

- **Normalização (2 FN):**
 - ✓ Se a disciplina BD não for oferecida para nenhuma turma, isso levará à remoção da terceira tupla da relação Turma, o que ocasionaria a perda do número de horas da disciplina BD, pois esta informação não se encontra em outra relação ou tupla.
 - ✓ Adicionalmente, outra anomalia é a duplicidade do número de horas das disciplinas, que pode levar à inconsistência (por exemplo, caso o número de horas da disciplina IP fosse atualizado na primeira tupla e não na segunda).
 - ✓ Observe que esses problemas são causados porque o atributo Num_horas depende apenas de parte da chave primária (no caso, depende da Sigla_Disciplina).

Normalização

- **Normalização (3 FN):**
 - ❖ Uma relação está na terceira forma normal quando duas condições são satisfeitas:
 - ✓ A relação está na segunda forma normal;
 - ✓ Não existir dependência transitiva na relação, ou seja, nenhum dos atributos não chave depende de outro atributo que também não é chave. Ou seja, se todos os atributos dependerem completa e exclusivamente da chave primária. Ou ainda, se não houver relação de identificação entre os atributos que não façam parte da chave primária (dependência transitiva).
 - ❖ Relações que estejam na 2FN e que não tenham nenhum ou apenas um atributo além da chave estão automaticamente na 3FN;

Normalização

- **Normalização (3 FN):**
 - ❖ Vejamos agora a seguinte relação:

Relação Funcionário

Cod_Empregado (PK)	Nome_Empregado	Cargo	Salário
100	Carlos Alves	Programador	2000
101	José Souza	Analista	3500
102	Maria Ramos	Programador	2000

- ❖ A relação Funcionário está na 2FN porque não contém chave composta.
- ❖ Porém, há uma dependência transitiva na mesma. O atributo Salário depende do atributo Cargo (que não faz parte da chave primária). Já o atributo Cargo, por sua vez, depende da chave primária Cod_Empregado. Assim, temos: $\text{Cod_Empregado} \rightarrow \text{Cargo} \rightarrow \text{Salário}$.

Normalização

Normalização (3 FN):

- ❖ **Para normalizar para a Terceira Forma Normal deve-se:**
 - ✓ Para passarmos uma entidade da 2FN para a 3FN, devemos eliminar as dependências transitivas:
 - a) Para cada atributo que não participe da chave primária da relação, analisar se existe dependência transitiva da chave primária. Para identificar a dependência transitiva de um atributo deve-se indagar: Qual outro atributo não pertencente à chave e poderia determinar o valor do atributo em análise? (você vai analisar se o valor de um atributo não-chave pode ser determinado por algum outro atributo que também não pertença à chave primária da relação);
 - b) Para cada grupo de atributos não-chaves dependentes funcionalmente de outros atributos não-chaves, cria-se uma nova relação. Essa nova relação vai ter os atributos dependentes como não-chaves e os atributos que causam a dependência (determinantes) como chave primária. O nome das novas relações deve ser escolhido de acordo com a chave primária das mesmas;

Normalização

Normalização (3 FN):

- ❖ **Para normalizar para a Terceira Forma Normal deve-se:**
 - ✓ Para passarmos uma entidade da 2FN para a 3FN, devemos eliminar as dependências transitivas:
 - c) Repetir os passos anteriores até que todos os atributos restantes na relação original dependam diretamente de toda a chave primária. Em outras palavras, repete-se até que todas as relações não contenham atributos dependentes de atributos não-chaves;
 - d) Excluir da tabela original todos os atributos com dependência transitiva, mantendo o atributo determinante da transitividade. Deve-se excluir, também, os atributos derivados a partir de outros (atributos calculados), como por exemplo, se houver um atributo data de nascimento e outro atributo idade, o atributo idade deve ser excluído, uma vez que, a qualquer momento, ele poderá ser calculado a partir do atributo data de nascimento.

Normalização

Normalização (3 FN):

- ❖ **Para normalizar para a Terceira Forma Normal deve-se:**
 - ✓ Por exemplo, para passar a relação Funcionário para a 3FN, iríamos criar uma nova relação (Salario_Cargo) que terá como chave primária o atributo não-chave determinante da dependência transitiva (Cargo) e o atributo não-chave dependente (Salário), uma vez que Cargo a Salário na relação Funcionário original. Depois, o atributo em dependência transitiva deve ser retirado da relação original (no caso, Salário é retirado da relação Funcionário). Assim, a relação original fica apenas com atributos que dependem funcionalmente apenas da chave primária (no caso, Cod_Empregado).

Normalização

- **Normalização (3 FN):**
 - ❖ **Para normalizar para a Terceira Forma Normal deve-se:**

Relação Funcionário

Cod_Empregado (PK)	Nome_Empregado	Cargo	Salário
100	Carlos Alves	Programador	2000
101	José Souza	Analista	3500
102	Maria Ramos	Programador	2000

Relação Funcionário (na 3FN)

Cod_Empregado (PK)	Nome_Empregado	Cargo
100	Carlos Alves	Programador
101	José Souza	Analista
102	Maria Ramos	Programador

Relação Salario_Cargo

Cargo (PK)	Salário
Programador	2000
Analista	3500

Normalização

- **Roteiro para Normalização:**
 - ❖ A partir do modelo relacional gerado a partir do modelo entidade relacionamento, construído na fase de modelagem conceitual, proceda os seguintes passos:
 - ✓ Elimine os atributos repetidos (se houver), de modo a obter um modelo na 1FN;
 - ✓ Elimine as dependências parciais da chave primária em suas relações (se houver) para obter o modelo na 2FN;
 - ✓ Elimine as dependências transitivas nas tabelas (se houverem), obtendo um esquema na 3FN;
 - ✓ Avalie se vale a pena aplicar mais algum tipo adicional de normalização FNBC e outras (4 e 5 FN).

Conteúdo Programático

UNIDADE III: ARQUITETURA E INFRAESTRUTURA DE DW

- Arquitetura de um DW: Componentes e suas características;
- Metadados: Gerência, Armazenamento e Integração.



Business School Brasil
O futuro é agora, o sucesso é garantido



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

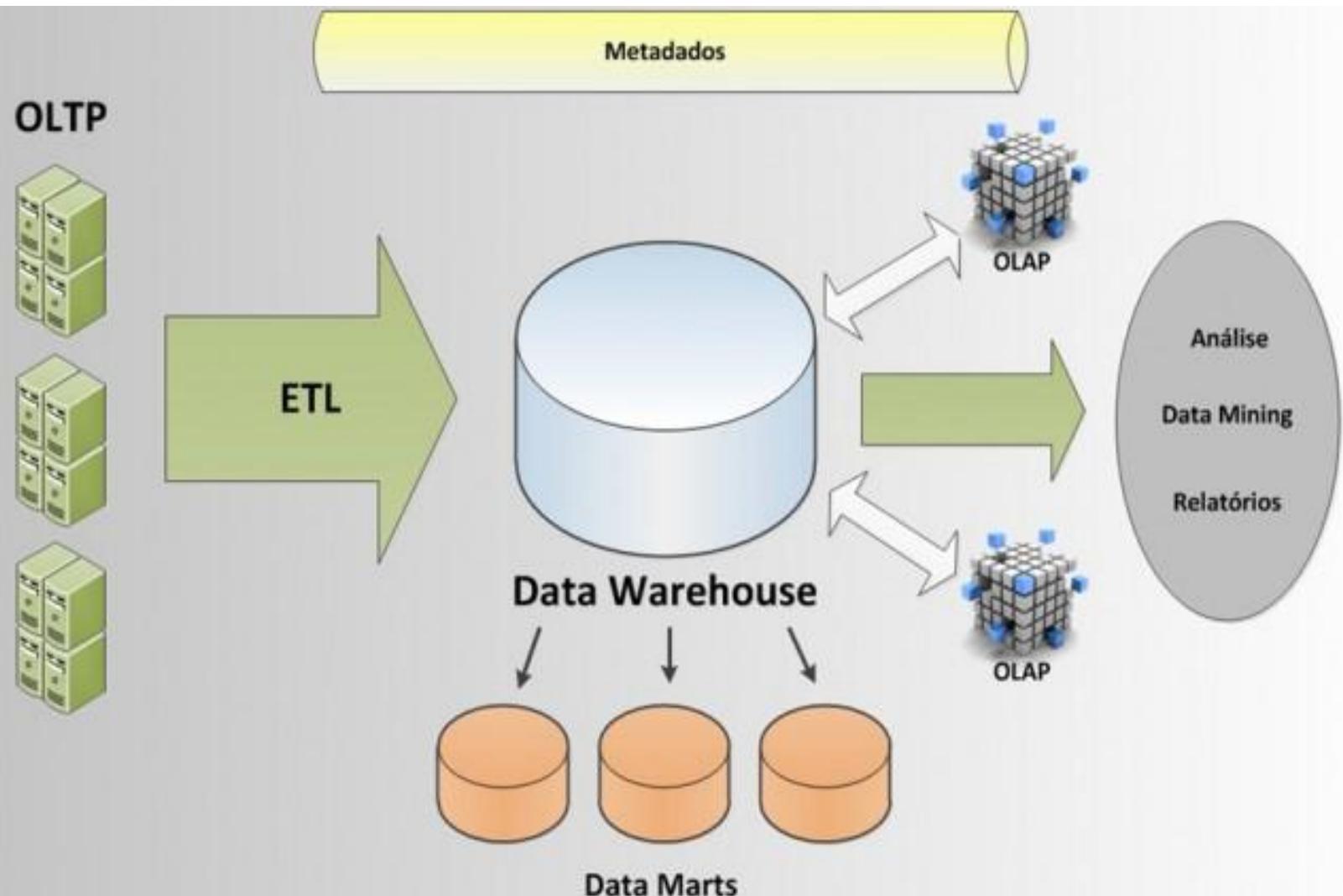
Arquitetura de DW

Visão superficial.



Arquitetura de DW

Arquitetura Genérica de um DW:



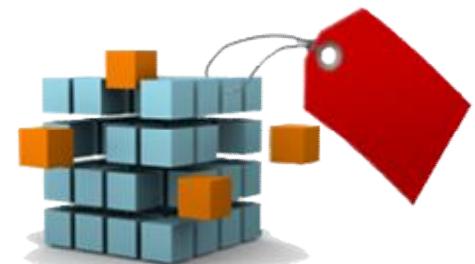
Arquitetura de DW

Opa! Temos alguns integrantes novos no Cenário DW

Metadados: Uma espécie de dicionário de dados, que agrega dados sobre outros dados, ou seja, dados que descrevem dados.

Os metadados são utilizados para facilitar o entendimento, o uso e o gerenciamento de dados. Eles variam conforme o tipo de dados e o contexto de uso.

Os metadados fornecem uma ligação essencial entre o gerador da informação e o usuário da mesma.



Arquitetura de DW

Opa! Temos alguns integrantes novos no Cenário DW

Data Mining: é o processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, criando novos subconjunto de dados.

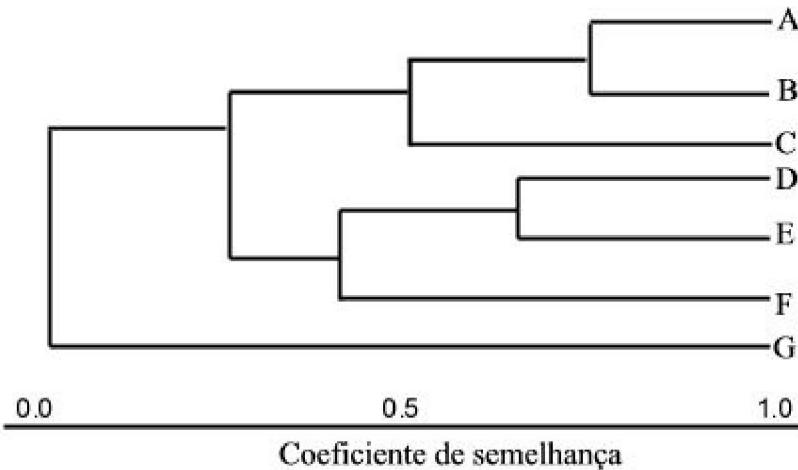
É formada por um conjunto de ferramentas e técnicas que através do uso de algoritmos de aprendizagem ou classificação.

São capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento.

Arquitetura de DW

Opa! Temos alguns integrantes novos no Cenário DW

O conhecimento adquirido pelo *Data Mining* pode ser apresentado por essas ferramentas de diversas formas: **agrupamentos, hipóteses, regras, árvores de decisão, grafos, ou dendogramas.**



Arquitetura de DW

Qual a diferença de BI com e sem *Data Mining*?

Sem DM o BI visa obter, a partir dos dados operativos brutos, informação útil para subsidiar a tomada de decisão nos escalões médios e altos da empresa;

Com DM o BI busca subsidiar a empresa com conhecimento novo e útil acerca do seu meio ambiente, servindo também ao plano estratégico.

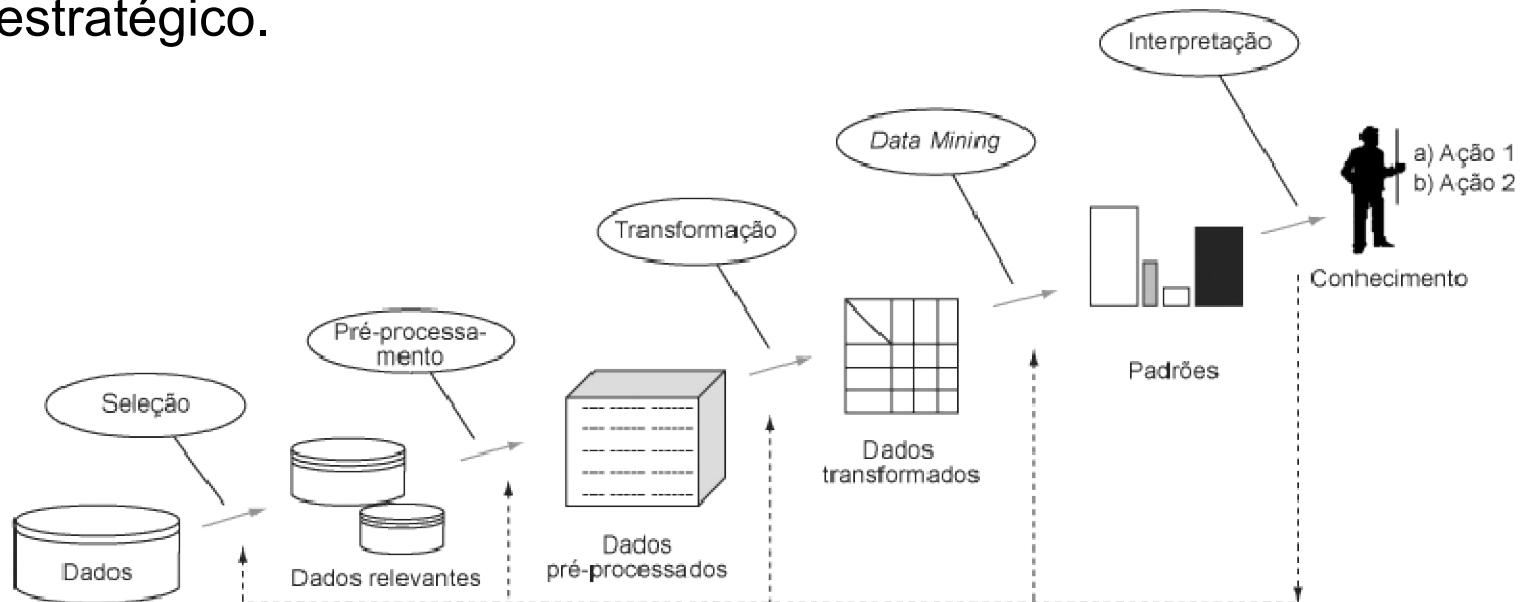


Figura 1. Etapas do processo KDD (Fayyad et al. (1996)).

Arquitetura de DW

Há alguma solução para se trabalhar com
Sistemas de Missão Crítica?



Business School Brasil
O futuro do negócio é educação



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ

Arquitetura de DW

ODS – *Operational Data Store*

É uma estrutura híbrida que tem características tanto de DW quanto de banco operacional.

Ao contrário de um banco de dados convencional os dados não retornam ao sistema de origem, mas podem ser utilizados em outras operações.

Características:

- Usados para decisões a curto prazo envolvendo aplicações de missão crítica;
- Trabalha diretamente com sistemas legados;
- Dados mais antigos podem ser movidos e summarizados para o DW.



Arquitetura de DW

ODS – *Operational Data Store*

- É uma estrutura híbrida que tem características tanto de DW quanto de banco operacional;
- O ODS é uma coleção de dados **orientada a assuntos, integrada, atual e volátil** utilizada para apoiar o processo de tomada de decisão tática na Empresa;
- Ao contrário de um banco de dados convencional os dados não retornam ao sistema de origem, mas podem ser utilizados em outras operações;
- **Características:**
 - ❖ Usados para decisões a curto prazo envolvendo aplicações de missão crítica;
 - ❖ Os seus dados podem ser movidos e summarizados no DW.

Arquitetura de DW

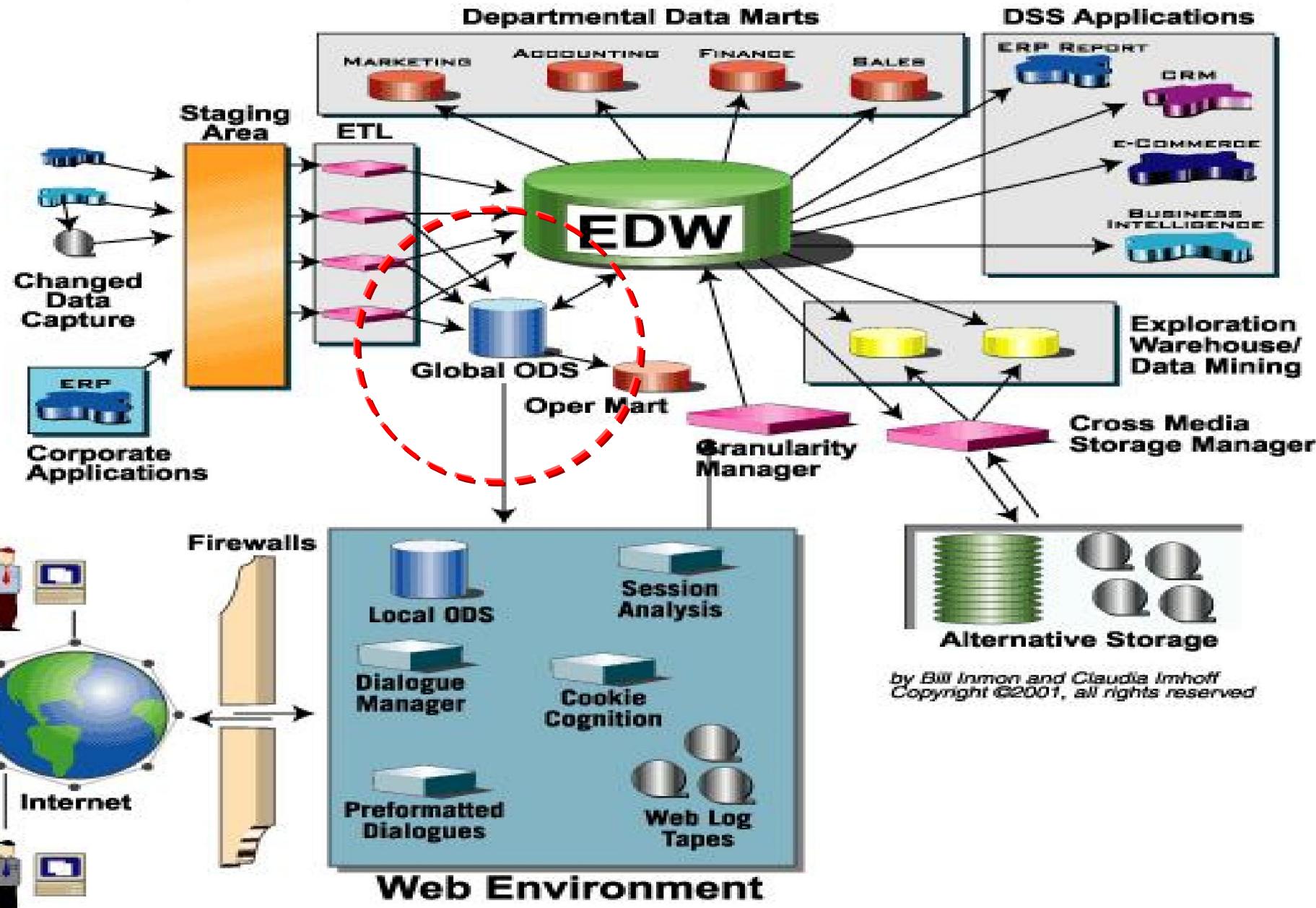
ODS – *Operational Data Store*

- Por ser uma estrutura híbrida ele é difícil de construir e de operar.
- Pontos importantes:
 - ❖ Ele tem tempo de resposta compatível com processamento OLTP (2/3 segundos);
 - ❖ Ele pode sofre atualizações constantes e periódicas (Capacidade de atualização on-line);
 - ❖ Ele é capaz de dar suporte a Sistemas de Suporte a Decisão (DSS).

Mas onde o ODS está situado dentro do *Data Warehousing*?

Arquitetura de DW

The Corporate Information Factory and the Web Environment



Arquitetura de DW

Como fica o desempenho dada a complexidade envolvida no *Data Warehousing*?



Business School Brasil

O futuro é agora. Agora é o seu.



PUC
GOIÁS



UNIVERSIDADE ESTADUAL
VALE DO ACARAÚ



Arquitetura de DW

Olhando o Desempenho:

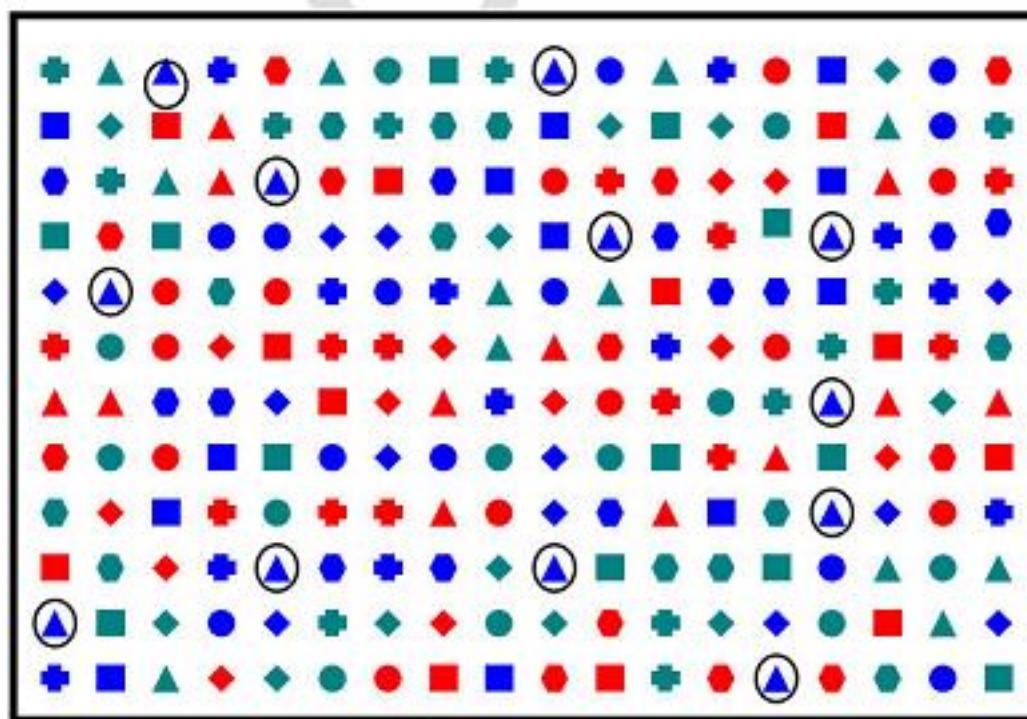
- Os DW crescem rapidamente com o passar do tempo, tornando-se necessário manter o desempenho do banco dentro de padrões aceitáveis.
- Podemos adotar algumas soluções como melhorar o hardware, computação em larga escala, junção de tabelas, etc.
- No entanto, podemos ter a necessidade de adotar soluções mais agressivas.



Arquitetura de DW

Olhando o Desempenho:

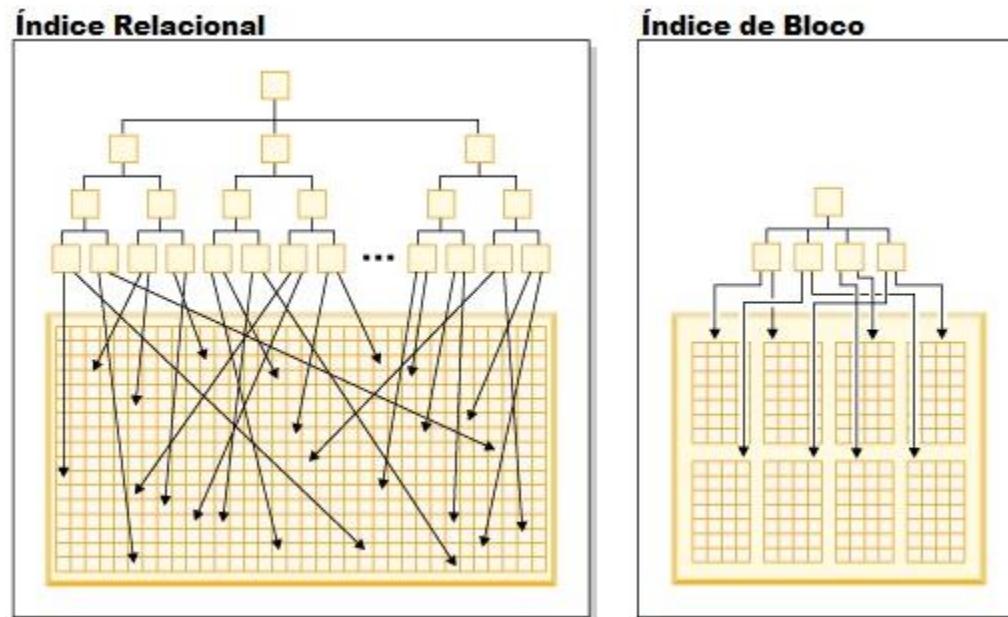
- Veja a tabela abaixo, onde os dados são figuras geométricas coloridas. Uma consulta neste banco para retornar triângulos azuis (estão destacados por círculos), varrerá a tabela inteira.



Arquitetura de DW

Olhando o Desempenho:

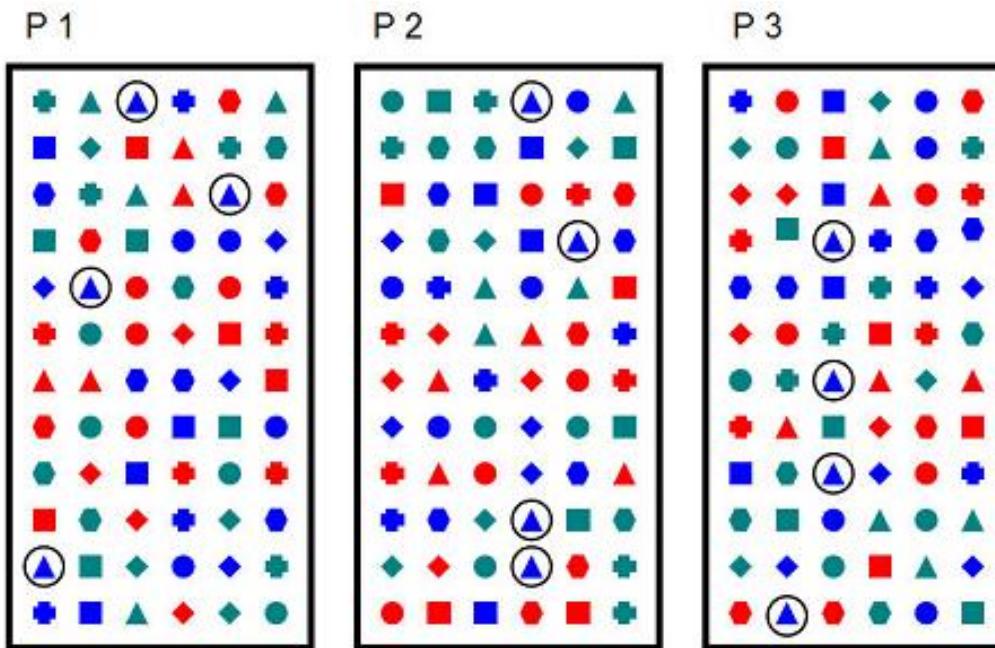
- Podemos adotar algumas soluções como:
 - a) **Índices**: fornecem o primeiro nível de melhoria de desempenho, mantendo os valores da coluna índice classificados e armazenados em uma estrutura de dados chamada árvores que contém o ID de linha (ou RID). Durante a recuperação, a árvore de índice é varrida em busca do valor procurado e, em seguida, todas as linhas relevantes apontados por RID são recuperadas. Isso evita a varredura de toda a tabela.



Arquitetura de DW

Olhando o Desempenho:

- Podemos adotar algumas soluções como:
b) Partição do Banco de Dados: Também conhecida como particionamento vertical. A Tabela é distribuída em formato de coluna. Podemos empregar paralelismo de maneira que cada partição seja checada simultaneamente.

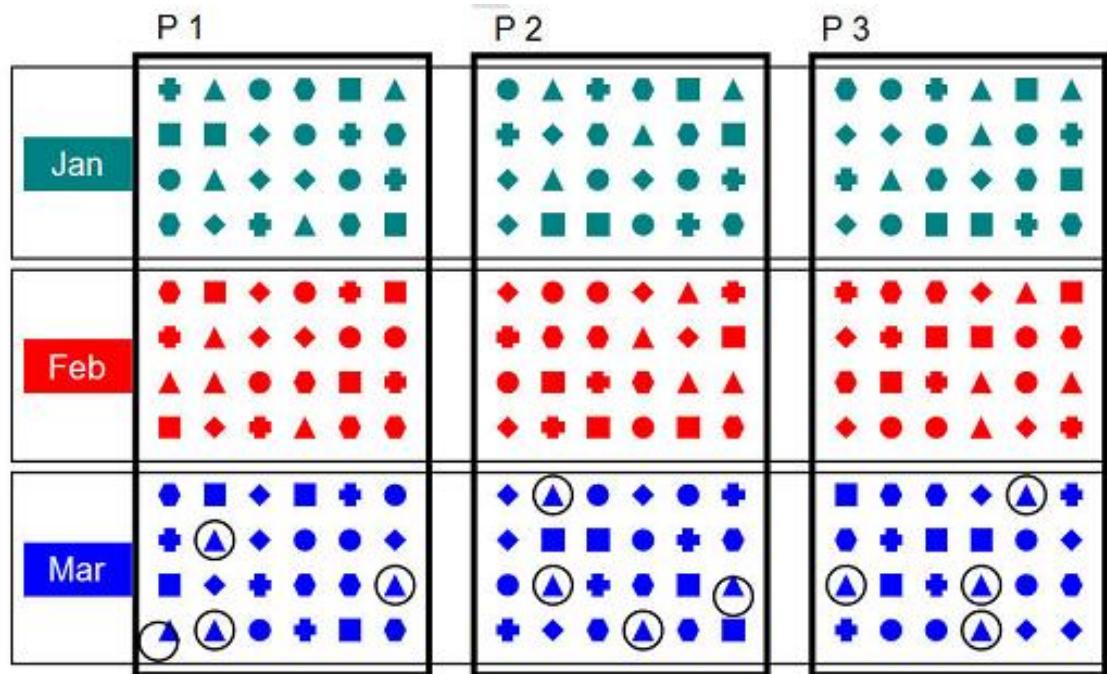


Ganho próximo a 3x

Arquitetura de DW

Olhando o Desempenho:

- Podemos adotar algumas soluções como:
c) Partição da Tabela: Também conhecida como particionamento horizontal. É um método de distribuição da tabela com base na divisão dos dados através de faixas de valores.



Ganho próximo a 9x.
3x da Partição do BD * 3x
da Partição da Tabela

Arquitetura de DW

Olhando o Desempenho:

- Podemos adotar algumas soluções como:

d) Clustering: Próximo nível de melhoria de desempenho é alcançado através de agrupamento de dados. *Clustering* significa armazenar dados semelhantes em conjunto no disco. Ao fazê-lo, as leituras seqüências no disco pode ser maximizada, o que proporciona melhor desempenho de leitura.

	P 1	P 2	P 3
Jan			
Feb			
Mar			

Agrupamento de dados pode ser feita em uma ou mais colunas da tabela.

Agrupamento de várias colunas também é denominado como *Multi Clustering Dimensional* ou MDC.

Olhando o Desempenho:

- Podemos adotar algumas soluções como:

e) Visões Materializadas: Uma visão materializada é uma visão com dados "armazenados". Ao contrário de uma visão normal (*view*), a visão materializada (*Materialized View*) armazena os resultados em um dispositivo de armazenamento permanente. Este conjunto de resultados é usado para todas as referências futuras em qualquer consulta que podem ser satisfeitas por meio subconjunto dos dados da visão materializada. Para melhorar ainda mais o tempo de consulta, as visões materializadas podem ser utilizadas em conjunto com qualquer uma das técnicas de melhoria de desempenho apresentadas anteriormente.