

# Gestão de Dados

Bem-vindo ao mundo da gestão de dados empresariais, onde informações estratégicas impulsionam decisões inteligentes e transformam negócios.

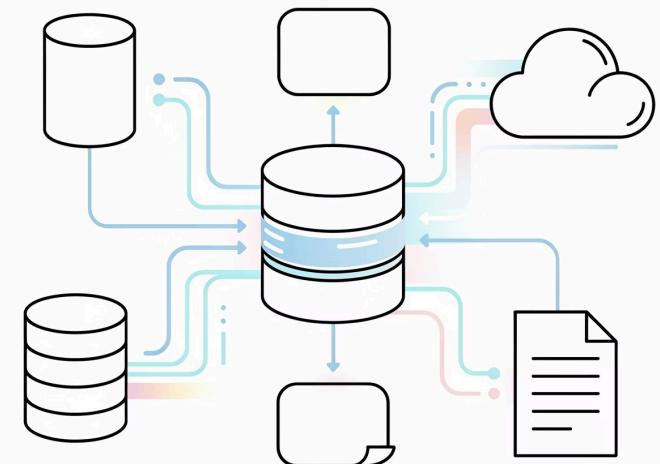
**Eduardo Ogasawara**

[eduardo.ogasawara@cefet-rj.br](mailto:eduardo.ogasawara@cefet-rj.br)  
<https://eic.cefet-rj.br/~eogasawara>

## Data Warehouse: A Base da Inteligência de Negócios

Um Data Warehouse é muito mais do que um simples repositório de dados. É uma arquitetura sofisticada que integra informações de múltiplas fontes, transformando dados brutos em insights estratégicos que impulsionam decisões empresariais inteligentes.

Esta tecnologia revolucionou a forma como as organizações gerenciam e analisam suas informações, permitindo análises históricas profundas e projeções futuras baseadas em dados consolidados e confiáveis.



## O Que É um Data Warehouse?

"Um data warehouse é uma coleção de dados orientada por assunto, integrada, variante no tempo e não volátil, para apoiar o processo de tomada de decisão da gerência." — W. H. Inmon

Embora existam diversas definições, o conceito central permanece consistente: um Data Warehouse é um banco de dados especializado para suporte à decisão, mantido separadamente dos sistemas operacionais da organização. Ele fornece uma plataforma sólida de dados históricos consolidados, especificamente projetada para análise e processamento de informações estratégicas.

O processo de **data warehousing** envolve a construção, manutenção e utilização efetiva desses repositórios, permitindo que organizações transformem dados em vantagens competitivas concretas.



## Orientado por Assunto



### Cliente

Dados demográficos, histórico de compras, preferências e comportamento de consumo



### Produto

Catálogo completo, especificações, desempenho de vendas e ciclo de vida



### Vendas

Transações, receitas, tendências de mercado e análise de performance

Um Data Warehouse é organizado em torno de temas principais do negócio, não de processos operacionais diários. Essa abordagem permite focar na modelagem e análise de dados para tomadores de decisão, proporcionando uma visão simples e concisa de questões específicas.

Ao contrário dos sistemas transacionais, o Data Warehouse exclui dados que não são úteis no processo de suporte à decisão, mantendo apenas informações relevantes e estratégicas para análises de negócios.

## Integrado: Unificando Fontes Heterogêneas

A construção de um Data Warehouse exige a integração de múltiplas fontes de dados heterogêneas, incluindo bancos de dados relacionais, arquivos planos e registros de transações online. Este processo complexo garante que informações dispersas sejam unificadas em um repositório coerente.

01

### Aplicação de Técnicas de Limpeza

Identificação e correção de inconsistências, erros e duplicações nos dados originais

02

### Padronização de Convenções

Garantia de consistência em nomenclaturas, estruturas de codificação e medidas de atributos

03

### Transformação e Conversão

Conversão de dados para o formato do warehouse durante o processo de migração

- Exemplo prático:** Preços de hotéis podem variar entre fontes de dados devido a diferenças em moeda, impostos inclusos, café da manhã coberto e estacionamento. O processo de integração resolve essas inconsistências.

## **Variante no Tempo: Perspectiva Histórica**

O horizonte temporal de um Data Warehouse é significativamente mais longo do que o dos sistemas operacionais, proporcionando uma dimensão histórica crucial para análises estratégicas.

### **Bancos de Dados Operacionais**

- Mantêm apenas dados atuais e recentes
- Focam em transações do momento presente
- Otimizados para operações em tempo real

### **Data Warehouse**

- Armazena informações históricas de 5 a 10 anos
- Permite análise de tendências de longo prazo
- Suporta previsões baseadas em padrões históricos

Toda estrutura de chave no Data Warehouse contém um elemento de tempo, seja explícita ou implicitamente. Esta característica permite rastreamento de mudanças ao longo do tempo e análises temporais sofisticadas, algo que nem sempre está presente nos dados operacionais.

## **Não Volátil: Estabilidade e Consistência**



### **Independência Física**

O Data Warehouse é um repositório fisicamente separado do ambiente operacional, contendo dados transformados e otimizados especificamente para análise. Esta separação garante que as operações analíticas não interfiram com o desempenho dos sistemas transacionais.

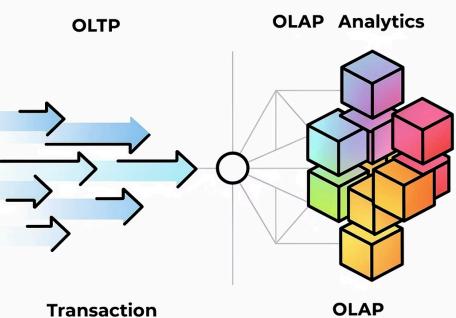
### **Natureza Estática**

Atualizações operacionais não ocorrem no ambiente do Data Warehouse. Uma vez que os dados são carregados, eles permanecem inalterados, garantindo consistência nas análises e relatórios ao longo do tempo. Esta estabilidade é fundamental para auditorias e comparações históricas.

### **Operações Simplificadas**

O ambiente não requer mecanismos complexos de processamento de transações, recuperação ou controle de concorrência. Apenas duas operações são necessárias: carga inicial dos dados e acesso para consultas analíticas, simplificando significativamente a arquitetura.

# OLTP vs. OLAP: Duas Abordagens Distintas



## OLTP

### Online Transactional Processing

Processamento transacional online focado em operações do SGBD, incluindo consultas rápidas e processamento de transações em tempo real para operações diárias do negócio.

- Otimizado para inserções, atualizações e exclusões
- Dados normalizados para eficiência operacional
- Resposta em milissegundos
- Suporta operações concorrentes massivas

## OLAP

### Online Analytical Processing

Processamento analítico online projetado para operações de Data Warehouse, incluindo análises multidimensionais como drilling, slicing e dicing de dados para insights estratégicos.

- Otimizado para consultas complexas e agregações
- Dados desnormalizados para análise rápida
- Resposta em segundos ou minutos
- Suporta análises históricas e preditivas

## Por Que um Data Warehouse Separado?

A separação entre sistemas operacionais e analíticos não é apenas uma preferência arquitetural, mas uma necessidade estratégica fundamentada em requisitos técnicos e de negócios distintos.



### Alto Desempenho para Ambos

SGBD otimizado para OLTP com métodos de acesso, indexação e controle de concorrência específicos. Warehouse ajustado para OLAP com suporte a consultas complexas, visões multidimensionais e consolidação de dados.



### Funções e Dados Diferentes

Suporte à decisão requer dados históricos que bancos operacionais normalmente não mantêm. Necessidade de consolidação através de agregação e sumarização de fontes heterogêneas para análises abrangentes.



### Qualidade e Consistência

Fontes distintas tipicamente usam representações, códigos e formatos de dados inconsistentes que precisam ser reconciliados. O Data Warehouse garante padrões unificados de qualidade.

- Nota importante:** Existem cada vez mais sistemas que realizam análises OLAP diretamente em bancos de dados relacionais, mas a arquitetura separada ainda oferece vantagens significativas para cenários empresariais complexos.

# Arquitetura Multi-Camadas do Data Warehouse



## Camada Inferior: Servidor do Data Warehouse

Fundação da arquitetura onde os dados são armazenados, gerenciados e organizados. Esta camada inclui o banco de dados do warehouse e os servidores que o suportam, garantindo armazenamento eficiente e acesso otimizado aos dados consolidados.



## Camada Intermediária: Servidor OLAP

Camada de processamento analítico que transforma dados brutos em insights açãoáveis. Implementa o motor de análise multidimensional, permitindo operações complexas como drill-down, roll-up, slice e dice sobre os dados do warehouse.

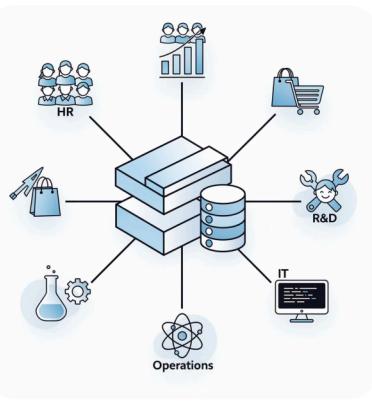


## Camada Superior: Ferramentas Front-End

Interface com o usuário final através de ferramentas de consulta, relatórios, análise, visualização e mineração de dados. Esta camada torna os insights acessíveis para tomadores de decisão através de dashboards intuitivos e relatórios personalizados.

Esta arquitetura em camadas garante separação de responsabilidades, escalabilidade e flexibilidade, permitindo que cada componente seja otimizado independentemente para sua função específica.

# Três Modelos de Data Warehouse



1

## Enterprise Warehouse

Coleta todas as informações sobre assuntos que abrangem toda a organização, fornecendo uma visão unificada e abrangente do negócio completo. É a solução mais robusta e complexa.

2

## Data Mart

Um subconjunto de dados corporativos valiosos para grupos específicos de usuários. Seu escopo é limitado a áreas selecionadas, como um data mart de marketing. Pode ser independente ou dependente do warehouse corporativo.

3

## Virtual Warehouse

Um conjunto de visões sobre bancos de dados operacionais. Apenas algumas das possíveis visões resumidas são materializadas, oferecendo uma alternativa leve e ágil, embora com limitações de desempenho.

A escolha do modelo depende de fatores como tamanho da organização, complexidade das necessidades analíticas, orçamento disponível e urgência da implementação. Muitas organizações começam com Data Marts e evoluem para um Enterprise Warehouse.

## Extração, Transformação e Carga (ETL)

O processo ETL é o coração operacional do Data Warehouse, responsável por mover, limpar e preparar dados de múltiplas fontes para análise. Este pipeline crítico garante que dados de qualidade alimentem o warehouse continuamente.



### Extração

Obtenção de dados de múltiplas fontes heterogêneas e externas



### Limpeza

Detecção de erros e correção quando possível



### Transformação

Conversão do formato legado para formato do warehouse



### Carga

Ordenação, consolidação e construção de índices



### Atualização

Propagação de updates das fontes para o warehouse

Durante a carga, múltiplas operações ocorrem simultaneamente: ordenação de dados, summarização de informações, consolidação de registros, computação de visões agregadas, verificação de integridade referencial, e construção de índices e partições para otimização de consultas. A fase de atualização mantém o warehouse sincronizado com as mudanças nas fontes operacionais.

# Repositório de Metadados: O Catálogo do Warehouse

Metadados são os dados que definem os objetos do warehouse, funcionando como um catálogo abrangente que documenta toda a estrutura, operação e uso do sistema. Este repositório é essencial para governança, manutenção e utilização efetiva do Data Warehouse.

## Estrutura do Warehouse

- Schema, visões e dimensões
- Hierarquias e dados derivados
- Localização e conteúdo dos data marts

## Metadados Operacionais

- Linhagem de dados e histórico de transformações
- Atualidade dos dados (ativo, arquivado ou purgado)
- Estatísticas de uso e relatórios de auditoria

## Algoritmos e Mapeamentos

- Algoritmos usados para summarização
- Mapeamento do ambiente operacional
- Transformações aplicadas aos dados

## Performance e Negócios

- Dados relacionados ao desempenho do sistema
- Termos e definições de negócios
- Propriedade de dados e políticas de cobrança



# Data Lakes

Explorando arquiteturas modernas de dados e suas aplicações no contexto empresarial

## Data Lake: Repositório Centralizado de Dados

Um data lake é um repositório centralizado que armazena todos os dados estruturados e não estruturados de uma organização, em qualquer escala. Esta abordagem revolucionária permite que as empresas mantenham dados em seu formato bruto, sem a necessidade de estruturação prévia.

Os dados são armazenados em seu estado original, permitindo a execução de diferentes tipos de análises — desde dashboards e visualizações até processamento de big data, análises em tempo real e machine learning. Esta flexibilidade orienta decisões empresariais mais informadas e estratégicas.

A capacidade de armazenar dados diversos sem transformação inicial reduz significativamente o tempo de ingestão e permite que as organizações preservem a granularidade e riqueza dos dados originais.



### Armazenamento Flexível

Dados em formato bruto



### Análises Diversas

Múltiplas aplicações



### Escalabilidade

Qualquer volume

## Arquitetura Conceitual de Data Lakes

A arquitetura de um data lake organiza o fluxo de dados desde a ingestão até o consumo analítico, garantindo governança e qualidade em todas as camadas do sistema.

### Organização de Dados Empresariais



#### Dados Não Estruturados

Documentos, imagens, vídeos e outros formatos livres

#### Extração

Processamento e parsing de dados brutos

#### Dados com Schema

Informações estruturadas e organizadas

#### Dados Estruturados

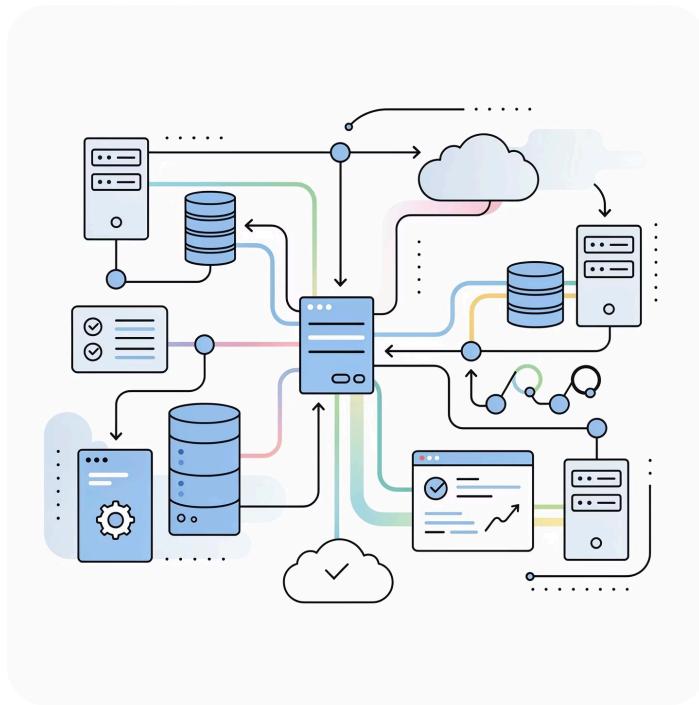
Tabelas relacionais prontas para análise

#### Analytics

Insights e visualizações de negócio

O data lake suporta todo este fluxo, desde a ingestão de dados brutos até a geração de análises sofisticadas que impulsionam decisões estratégicas.

## Desafios na Implementação de Data Lakes



### Volume Massivo

Centenas de milhares de conjuntos de dados ou mais, exigindo gerenciamento sofisticado

### Consultas Complexas

Queries por palavra-chave, descoberta de datasets relacionáveis, identificação de features relevantes

### Gestão de Metadados

Tarefa crítica na construção de data lakes eficientes e governados

O sucesso de um data lake depende fundamentalmente da capacidade de gerenciar metadados de forma eficaz, permitindo descoberta, governança e qualidade de dados em escala empresarial.

# Diversidade e Integração em Data Lakes

Os formatos de datasets no mundo real podem ser altamente heterogêneos, apresentando desafios únicos de integração e processamento.

01

## Ingestão e Extração

Processo de trazer datasets estruturados para o data lake, incluindo ingestão de dados já estruturados e extração de dados estruturados de fontes não estruturadas e semi-estruturadas

02

## Integração de Dados

Tarefa de encontrar tabelas que podem ser unidas (join) ou combinadas (union), ou de popular esquemas sob demanda com todos os dados do lake que conformam ao schema definido

- Exemplo Prático:** Enriquecer Registros Eletrônicos de Saúde (EPR) usando dados de diversos conjuntos de dados pessoais de saúde não padronizados para melhor prever riscos à saúde. Isso envolve unir ou combinar tabelas de diferentes datasets e fontes heterogêneas.

## Tarefas Comuns na Construção de Data Lakes



### Ingestão

Captura e carregamento de dados de múltiplas fontes



### Extração

Inferência de tipos e estruturação de dados brutos



### Gestão de Metadados

Catalogação e governança de ativos de dados



### Limpeza

Tratamento de inconsistências e qualidade



### Integração

Combinação de datasets relacionados



### Descoberta

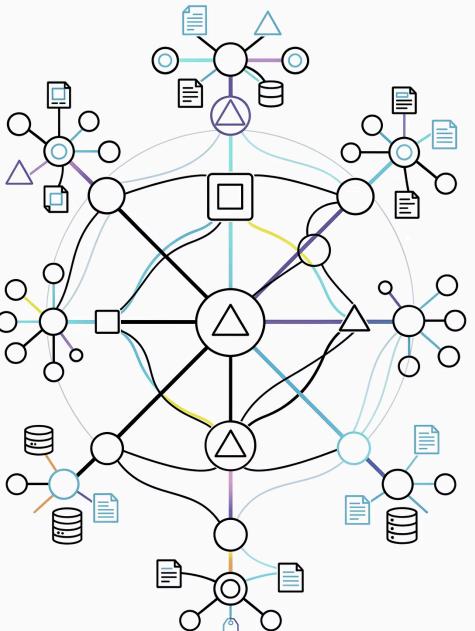
Localização de dados relevantes



### Versionamento

Controle de evolução e histórico de dados

Estas sete tarefas formam o núcleo operacional de qualquer implementação bem-sucedida de data lake, garantindo dados acessíveis, confiáveis e açãoáveis.



## Gestão de Metadados: Estratégias Avançadas

### Enriquecimento Semântico

Enriquecer dados e metadados com informação semântica, suportando consultas baseadas em templates sobre metadados

### Extração Profunda

Extrair metadados profundamente embutidos e metadados contextuais para suportar descoberta baseada em tópicos

### Integração Visual

Integrar metadados sobre dados, usuários e consultas, suportando análises visuais e interativas

A gestão eficaz de metadados transforma um data lake de um repositório passivo em um ativo inteligente que facilita descoberta, governança e extração de valor dos dados.

## Dois Propósitos e Ecossistemas Distintos

### Data Warehouses



### Data Lakes



### Casos de Uso Analíticos

Suporte à decisão baseado em dados estruturados

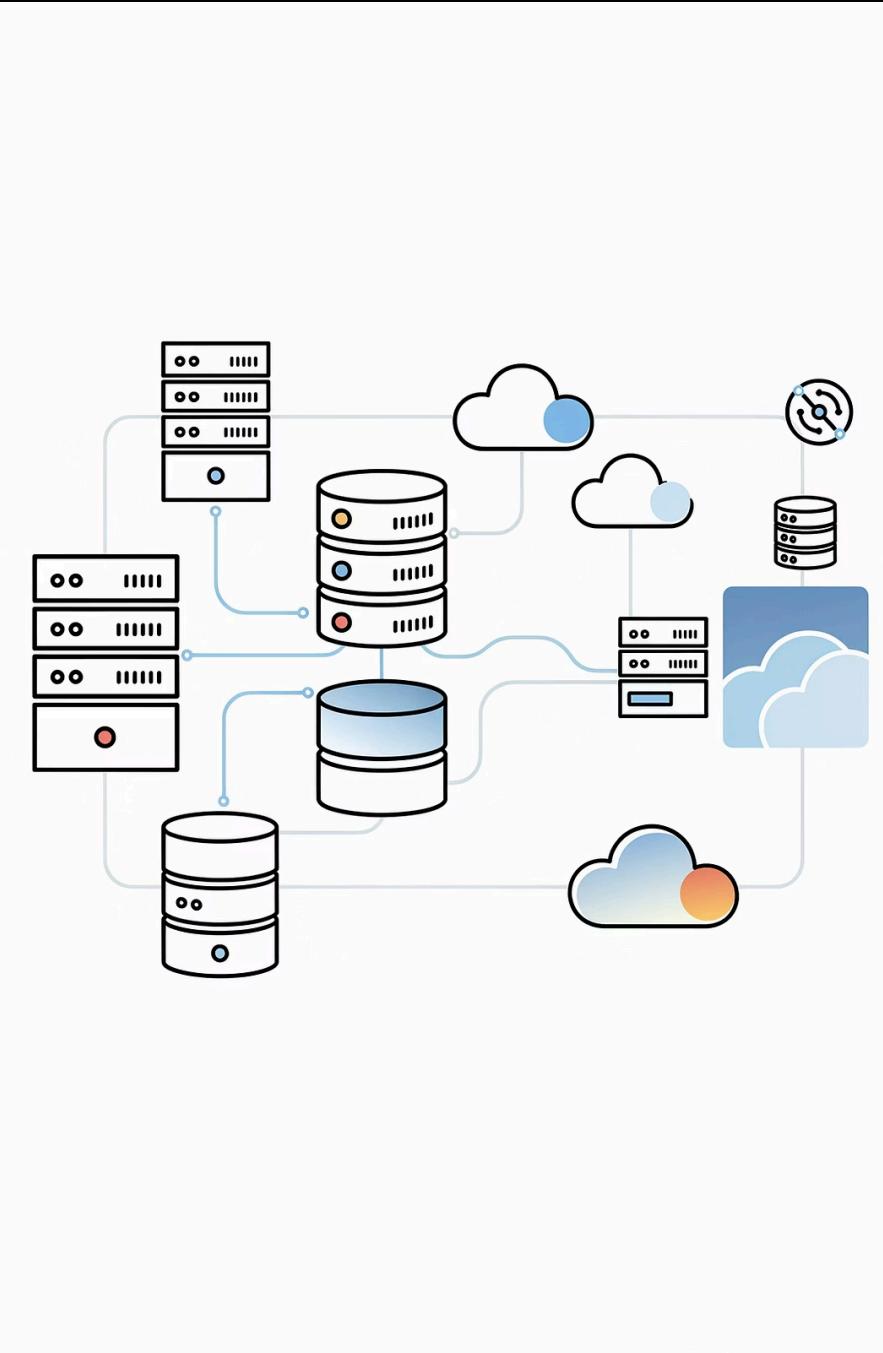
- Dados estruturados e organizados
- SQL e Python como linguagens principais
- Foco em relatórios e BI tradicional
- Schemas definidos e otimizados

### Casos de Uso Operacionais

Inteligência de dados em aplicações

- Dados brutos e diversos formatos
- Java, Scala, Python, R e SQL
- Machine learning e analytics avançados
- Flexibilidade e experimentação

**Convergência Emergente:** As fronteiras entre data warehouses e data lakes estão se tornando cada vez mais fluidas, com arquiteturas modernas buscando combinar os melhores aspectos de ambas as abordagens.



# Arquiteturas Modernas

## Data Lakehouses: O Melhor de Dois Mundos



"Um data lakehouse é uma nova arquitetura aberta de gerenciamento de dados que combina a flexibilidade, custo-eficiência e escala de data lakes com o gerenciamento de dados e transações ACID de data warehouses, permitindo business intelligence (BI) e machine learning (ML) sobre todos os dados."

— Databricks

O conceito de data lakehouse representa uma evolução significativa na arquitetura de dados, eliminando a necessidade de manter sistemas separados para diferentes cargas de trabalho.

Esta abordagem unificada reduz a complexidade operacional, minimiza a duplicação de dados e permite que as organizações obtenham insights mais rapidamente, mantendo a governança e qualidade necessárias para decisões críticas de negócios.

1

**Arquitetura Unificada**

Sistema único para múltiplas  
cargas

**50%**

**Redução de Custos**

Menos duplicação de dados

## Características Fundamentais dos Lakehouses



### Cargas de Trabalho Diversas

- Data science e experimentação
- Machine learning em produção
- SQL e analytics tradicionais
- Processamento de lotes e streams

### Streaming End-to-End

Capacidade de processar dados em tempo real e gerar relatórios instantâneos, eliminando latências tradicionais entre ingestão e análise.

# Data Fabric e Virtualização de Dados

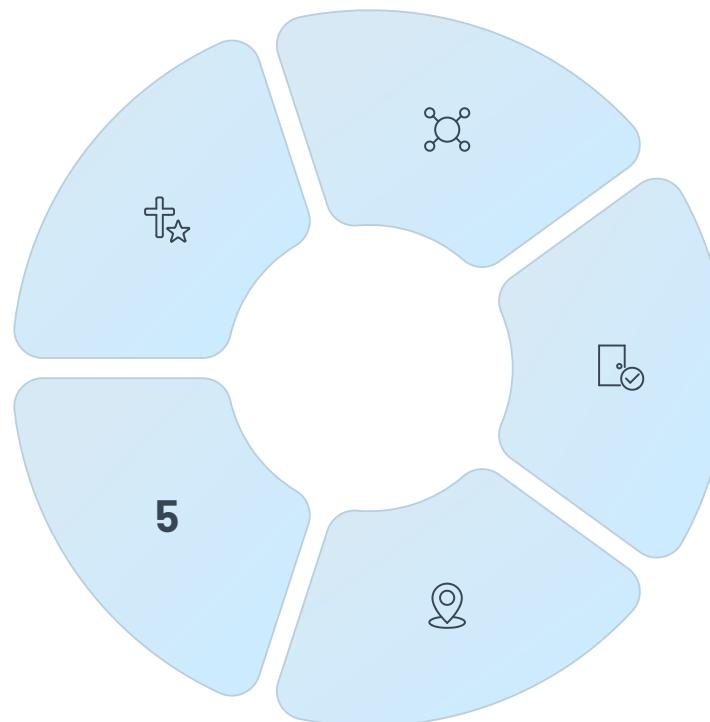
"Data fabric é uma arquitetura que facilita a integração end-to-end de vários pipelines de dados e ambientes de nuvem através do uso de sistemas inteligentes e automatizados"

— IBM

## Arquitetura Holística

Gerenciamento unificado de dados em todo o ecossistema

Um data fabric fornece uma camada de abstração inteligente que permite às organizações tratar dados distribuídos como se fossem um único repositório lógico, simplificando significativamente a governança e o consumo de dados.



## Abstração de Complexidade

Oculta heterogeneidade subjacente

## Ponto Único de Acesso

Acesso consolidado a todos os dados

## Independência de Localização

Dados acessíveis independente de fonte, formato ou tecnologia

## Data Mesh: Arquitetura Descentralizada

Um data mesh cria múltiplos sistemas específicos por domínio, cada um especializado de acordo com suas funções e usos, trazendo assim os dados para mais perto dos consumidores.

Esta abordagem fundamentalmente diferente trata dados como um produto, com equipes de domínio assumindo propriedade completa de seus dados - desde a qualidade até a entrega.



**Data Mesh é um padrão arquitetural específico focado no gerenciamento de dados**, enfatizando descentralização, ownership de domínio, e tratamento de dados como produtos de primeira classe.

# Virtualização de Dados: Componente Essencial

"Virtualização de dados é uma das tecnologias que habilita uma abordagem de data fabric. Em vez de mover fisicamente os dados de várias fontes on-premises e na nuvem usando os processos tradicionais de ETL (extract, transform, load), uma ferramenta de virtualização de dados conecta-se às diferentes fontes, integrando apenas os metadados necessários e criando uma camada virtual de dados"

— *Definição Técnica*

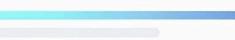
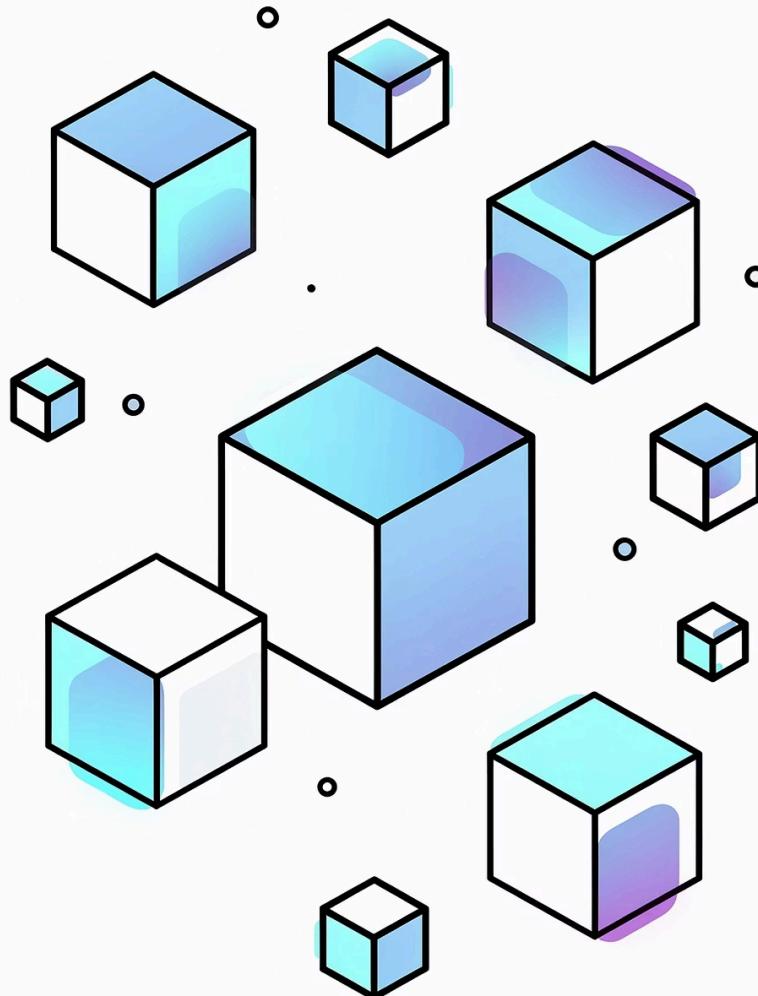
## Benefícios da Virtualização

- Elimina necessidade de movimentação física de dados
- Reduz latência no acesso a informações
- Diminui custos de armazenamento e ETL
- Fornece visão unificada em tempo real
- Simplifica manutenção e governança

## Integração Arquitetural

A virtualização de dados pode ser um componente ou camada em uma arquitetura de data fabric, fornecendo a abstração necessária para acessar dados distribuídos sem replicação física.

Esta tecnologia é fundamental para organizações que precisam de acesso ágil a dados distribuídos mantendo fonte única de verdade.



# Modelagem Dimensional de Data Warehouses

Uma introdução completa aos conceitos e práticas fundamentais para a construção de estruturas multidimensionais eficientes

## Das Tabelas e Planilhas aos Cubos de Dados

Um data warehouse é fundamentado em um modelo de dados multidimensional que visualiza informações na forma de um cubo de dados. Esta estrutura revolucionária permite análises complexas e eficientes de grandes volumes de dados empresariais.

Um cubo de dados, como o de vendas, possibilita modelar e visualizar dados em múltiplas dimensões simultaneamente. Por exemplo, você pode analisar vendas por produto, região e período temporal ao mesmo tempo.

01

### Base Cuboid

O cubo base n-dimensional que contém o nível mais detalhado de dados

02

### Apex Cuboid

O cuboide 0-dimensional no topo, com o mais alto nível de sumarização

03

### Lattice de Cuboides

A estrutura completa que forma o cubo de dados multidimensional

### Componentes Principais

- **Tabelas de dimensão:** item (nome, marca, tipo) ou tempo (dia, semana, mês, trimestre, ano)
- **Tabela fato:** contém medidas (como valor\_vendido) e chaves para cada tabela de dimensão relacionada
- **Cuboides:** formam a estrutura em rede do cubo de dados

## Modelagem Conceitual de Data Warehouses

A modelagem dimensional é o coração do data warehouse, definindo como dimensões e medidas se organizam para suportar análises complexas. Três esquemas principais dominam esta arquitetura, cada um com características e aplicações específicas.



### Star Schema

Uma tabela fato central conectada a um conjunto de tabelas de dimensão, formando uma estrutura em estrela simples e eficiente

- Estrutura desnortinalizada
- Consultas mais rápidas
- Fácil compreensão



### Snowflake Schema

Um refinamento do star schema onde hierarquias dimensionais são normalizadas em tabelas menores, criando forma similar a um floco de neve

- Estrutura normalizada
- Economia de espaço
- Menor redundância



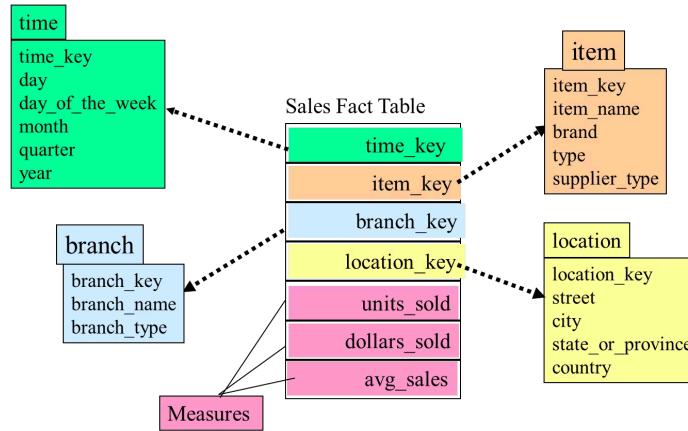
### Fact Constellation

Múltiplas tabelas fato compartilham tabelas de dimensão, visto como coleção de estrelas (também chamado galaxy schema)

- Múltiplos processos de negócio
- Dimensões compartilhadas
- Análises integradas

## EXEMPLO PRÁTICO

# Star Schema: Um Exemplo Detalhado



## Tabela Fato de Vendas

A tabela central que armazena as métricas de negócio e as chaves estrangeiras para as dimensões.

### Chaves de Dimensão:

- time\_key (chave temporal)
- item\_key (chave de produto)
- branch\_key (chave de filial)
- 
- location\_key (chave de localização)

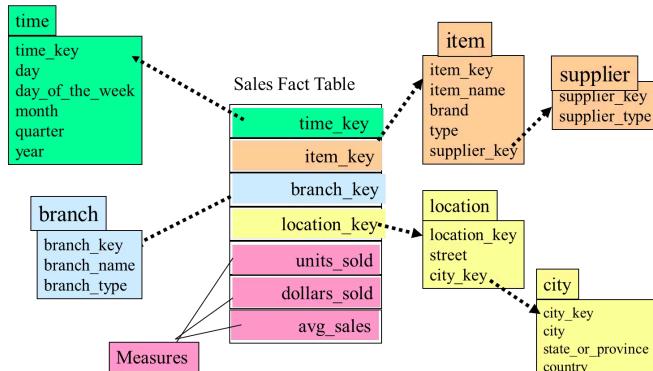
### Medidas:

- units\_sold (unidades vendidas)
- dollars\_sold (valor vendido)
- avg\_sales (média de vendas)

- **Vantagem Principal:** No star schema, cada dimensão está diretamente conectada à tabela fato, resultando em consultas SQL simples com menos JOINs e melhor performance em queries analíticas.

# Snowflake Schema: Estrutura Normalizada

O snowflake schema representa uma evolução do star schema através da normalização das tabelas de dimensão. Neste modelo, as hierarquias dentro das dimensões são decompostas em múltiplas tabelas relacionadas, criando uma estrutura ramificada que lembra um floco de neve.



## Tabela Fato Central

Mantém as mesmas chaves e medidas: time\_key, item\_key, branch\_key, location\_key, units\_sold, dollars\_sold, avg\_sales

## Dimensões Normalizadas

Cada dimensão é decomposta em hierarquias, com tabelas separadas para cada nível (ex: item → categoria → departamento)

## Relacionamentos em Cascata

As tabelas de dimensão conectam-se entre si, formando cadeias de relacionamentos que refletem a hierarquia natural dos dados

## Vantagens

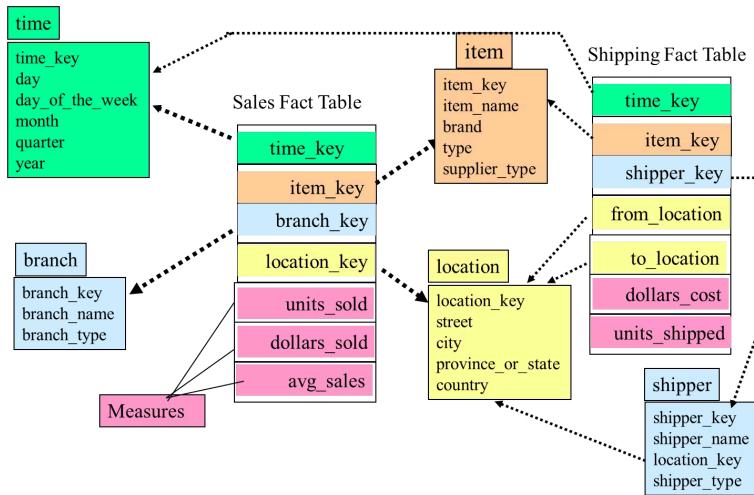
- Menor redundância de dados
- Economia de espaço de armazenamento
- Facilita manutenção de hierarquias
- Integridade referencial mais forte

## Desvantagens

- Consultas mais complexas
- Maior número de JOINs necessários
- Performance pode ser menor em queries
- Mais difícil para usuários entenderem

## Fact Constellation: Múltiplas Perspectivas de Análise

O fact constellation, também conhecido como galaxy schema, representa o modelo mais complexo e flexível de data warehouse. Ele permite que múltiplas tabelas fato compartilhem dimensões comuns, possibilitando análises integradas de diferentes processos de negócio.



### Tabela Fato de Vendas

#### Chaves:

- time\_key
- item\_key
- branch\_key
- location\_key

#### Medidas:

- units\_sold
- dollars\_sold
- avg\_sales

### Tabela Fato de Envios

#### Chaves:

- time\_key
- item\_key
- shipper\_key
- from\_location
- to\_location

#### Medidas:

- dollars\_cost
- units\_shipped

- Aplicação Prática:** Este modelo é ideal quando você precisa analisar múltiplos processos de negócio que compartilham dimensões comuns, como vendas e envios que compartilham as dimensões de tempo, item e localização.

## HIERARQUIAS

### Hierarquia de Conceitos para Dimensões

As dimensões em um data warehouse não são planas - elas contêm hierarquias naturais que permitem análises em diferentes níveis de granularidade. Compreender estas hierarquias é essencial para realizar operações de drill-down e roll-up eficazmente.

#### Exemplo: Hierarquia da Dimensão Localização



##### All (Todos)

Nível mais agregado - visão global de todos os dados



##### Region (Região)

Europa, América do Norte, Ásia, etc.



##### Country (País)

Alemanha, Espanha, Canadá, México, etc.



##### City (Cidade)

Frankfurt, Vancouver, Toronto, etc.



##### Office (Escritório)

Nível mais detalhado - L. Chan, M. Wind, etc.

Esta estrutura hierárquica permite que analistas naveguem pelos dados em diferentes níveis de detalhe, desde uma visão global até informações específicas de cada escritório individual. Cada nível da hierarquia agrupa os dados dos níveis inferiores, facilitando análises tanto estratégicas quanto operacionais.

## Categorias de Medidas em Cubos de Dados

As funções de agregação em cubos de dados podem ser classificadas em três categorias fundamentais, cada uma com propriedades matemáticas específicas que afetam diretamente o desempenho e a implementação do data warehouse.

### Distributivas

O resultado obtido aplicando a função a n valores agregados é igual ao resultado aplicado a todos os dados sem particionamento.

#### Exemplos:

- `count()` - contagem de registros
- `sum()` - soma de valores
- `min()` - valor mínimo
- `max()` - valor máximo

**Vantagem:** Mais eficientes para pré-computação em cubos de dados

### Algébricas

Podem ser computadas por uma função algébrica com M argumentos (onde M é um inteiro limitado), cada um obtido aplicando funções distributivas.

#### Exemplos:

- $\text{avg}(x) = \text{sum}(x) / \text{count}(x)$
- `standard_deviation()`
- `variance()`

**Questão:** `min_N()` é uma medida algébrica?

### Holísticas

Não existe limite constante no tamanho de armazenamento necessário para descrever um subaggregado.

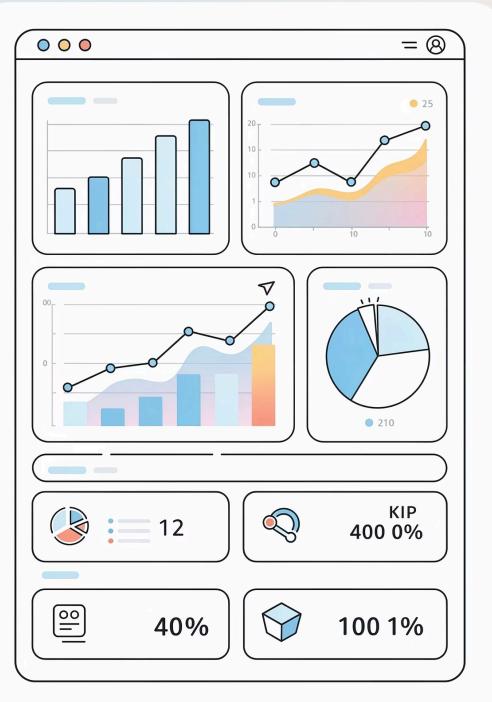
#### Exemplos:

- `median()` - mediana
- `mode()` - moda
- `rank()` - classificação

**Desafio:** Requerem dados completos para cálculo preciso, dificultando pré-agregação

# Fundamentos Conceituais do OLAP

Online Analytical Processing (OLAP) representa a tecnologia central que permite análises multidimensionais eficientes em data warehouses. É a ponte entre dados armazenados e insights de negócios açãoáveis.



## Visão Multidimensional

OLAP permite visualizar dados através de múltiplas dimensões simultaneamente, facilitando a compreensão de relacionamentos complexos e padrões ocultos nos dados empresariais.



## Análise Rápida

Estruturas otimizadas e pré-agregações permitem respostas em tempo quase real para consultas complexas envolvendo milhões de registros.



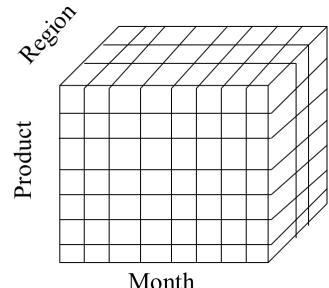
## Navegação Intuitiva

Operações como drill-down, roll-up, slice e dice permitem que analistas explorem dados naturalmente, seguindo o raciocínio analítico.

## ESTRUTURA MULTIDIMENSIONAL

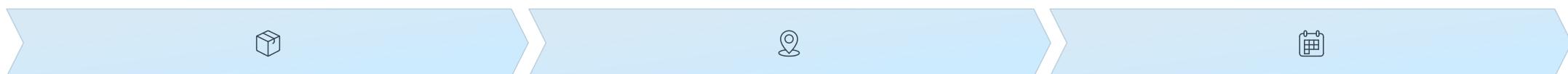
### Dados Multidimensionais em Ação

Considere o volume de vendas como uma função de produto, mês e região. Este exemplo ilustra como múltiplas dimensões se combinam para criar uma estrutura analítica poderosa e flexível.



#### Dimensões: Produto, Localização, Tempo

Cada dimensão possui hierarquias naturais que permitem agregações em diferentes níveis de detalhe.



#### Produto

**Industry** → Category → Product

Do nível mais agregado (indústria) até produtos individuais

#### Localização

**Region** → Country → City → Office

Da visão regional até escritórios específicos

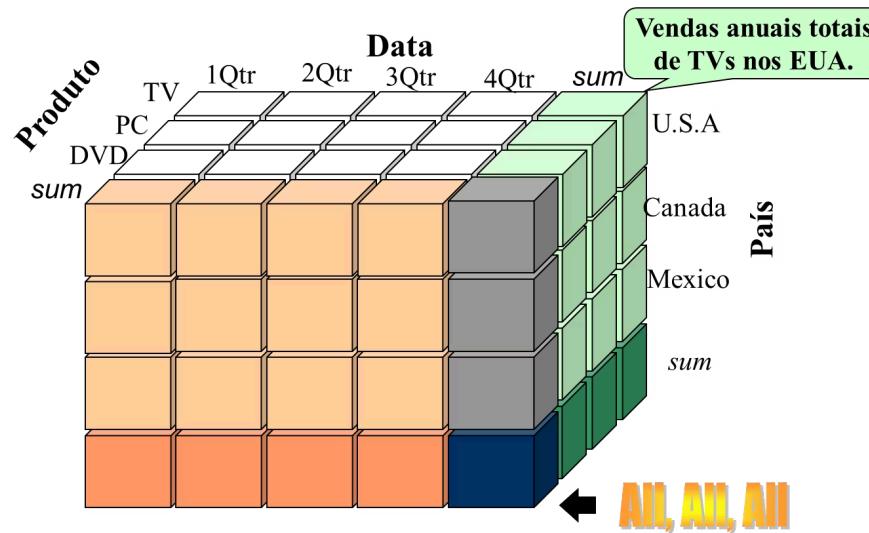
#### Tempo

**Year** → Quarter → Month → Week → Day

De anos completos até dias individuais

- Exemplo Prático: Um gerente pode iniciar analisando vendas anuais por região, fazer drill-down para vendas mensais em países específicos, e finalmente examinar vendas diárias de produtos individuais em escritórios particulares.

## Exemplo de Cubo de Dados



Este exemplo demonstra um ponto específico no cubo de dados tridimensional, representando as vendas totais anuais de televisores nos EUA. É uma agregação que combina:

- **Dimensão Produto:** TV (categoria específica)
- **Dimensão Localização:** U.S.A. (país específico)
- **Dimensão Tempo:** Year (nível anual)

Esta célula do cubo contém a medida agregada (vendas totais) para esta combinação específica de dimensões.

### Seleção de Dimensões

Escolha dos atributos relevantes para análise (produto, local, tempo)

### Definição de Granularidade

Seleção do nível de detalhe para cada dimensão (TV, USA, Year)

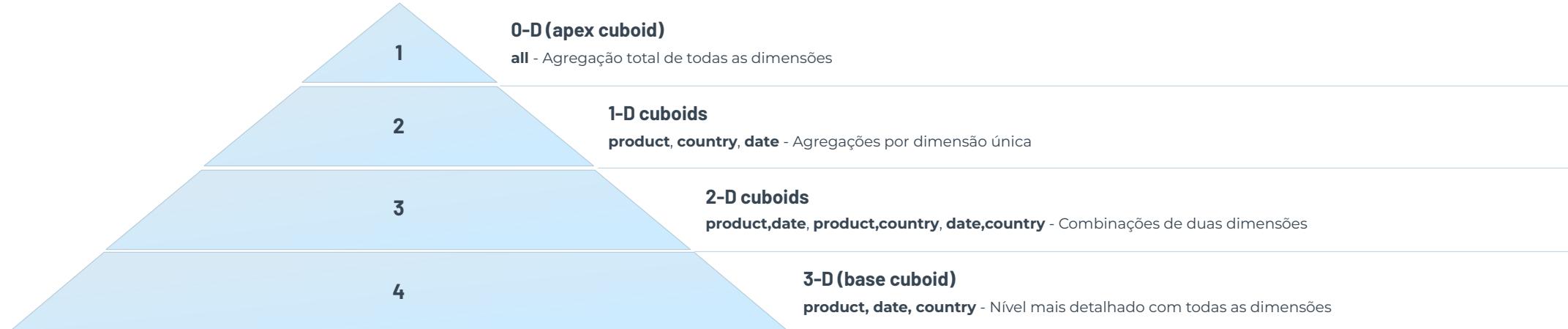
### Agregação de Medidas

Cálculo das métricas de negócios (vendas totais anuais)

## ESTRUTURA DE CUBOIDES

### Cuboides Correspondentes ao Cubo

Um cubo de dados completo é composto por múltiplos cuboides, cada um representando uma combinação diferente de dimensões. Esta estrutura hierárquica permite agregações eficientes em todos os níveis.



#### Apex Cuboid

O cuboide de nível mais alto contém apenas uma célula com a agregação total de todas as medidas. É o ponto de partida para análises top-down.

1

#### Apex (0-D)

Um único cuboide com total geral

3

#### Cuboídes 1-D

Três cuboídes com agregação por dimensão

3

#### Cuboídes 2-D

Três cuboídes com pares de dimensões

1

#### Base (3-D)

Um cuboide com máximo detalhe

## Online Analytical Processing (OLAP)

OLAP representa a tecnologia que permite explorar sistematicamente todos os possíveis subespaços de um cubo de dados em busca de padrões interessantes e insights valiosos para o negócio.

### 1 Exploração Conceitual

Conceitualmente, podemos explorar todos os possíveis subespaços de dados em busca de padrões interessantes, mas surgem questões fundamentais sobre eficiência e relevância.

### 2 Problemas Fundamentais

**Quais padrões são interessantes?** Como definir relevância e valor em meio a milhares de possíveis agregações?

**Como explorar sistematicamente?** Métodos eficientes para navegar todos os subespaços sem computação excessiva.

### 3 Importância de Agregações

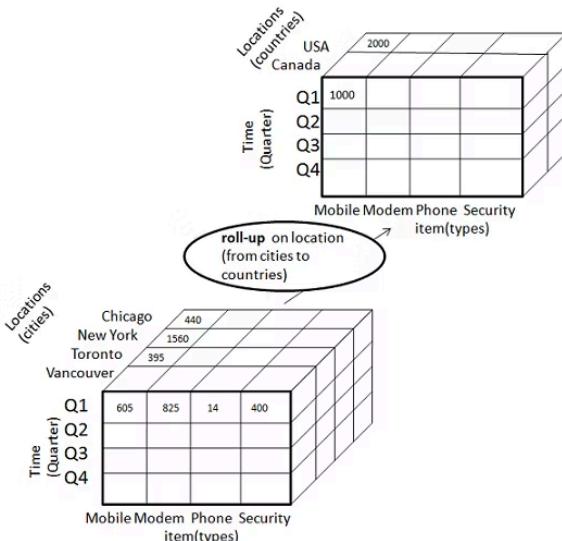
Agregações e group-bys são ferramentas essenciais em análise e sumarização de dados. No benchmark TPC, das 83 queries padrão, agregações aparecem 59 vezes e group-bys 20 vezes.

```
SELECT time, altitude, AVG(temp)
FROM weather
GROUP BY time, altitude;
```

OLAP fornece as técnicas que respondem queries analíticas multidimensionais (MDA) de forma eficiente, permitindo que analistas de negócios obtenham insights rapidamente sem necessidade de conhecimento técnico profundo em SQL ou estruturas de banco de dados.

## Operações OLAP Fundamentais

As operações OLAP permitem que analistas naveguem pelos dados em diferentes níveis de granularidade, explorando relacionamentos e descobrindo insights através de transformações sistemáticas do cubo de dados.



### Roll Up (Drill-Up)

Sumariza dados subindo na hierarquia ou através de redução dimensional. Move de dados detalhados para agregações mais amplas.

### Exemplo de Transformação:

(Day, Store, Product type, SUM(sales))

↓

(Month, City, \*, SUM(sales))

- Day → Month (agregação temporal)
- Store → City (agregação espacial)
- Product type → \* (remoção de dimensão)

### Drill Down (Roll-Down)

O reverso do roll-up, movendo de sumarizações de alto nível para dados mais detalhados, ou introduzindo novas dimensões na análise.

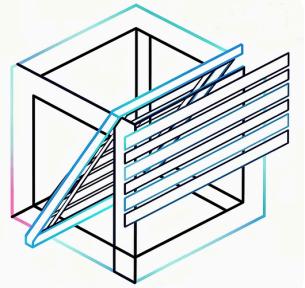
### Tipos de Drill-Down:

- **Descida hierárquica:** Month → Week → Day
- **Aumento de granularidade:** City → Store → Terminal
- **Adição de dimensões:** Incluir região ou categoria previamente agregada

- Caso de Uso:** Um executivo pode começar visualizando vendas mensais totais por cidade (roll-up), identificar uma anomalia, e então fazer drill-down para examinar vendas diárias por loja específica e categoria de produto para investigar a causa raiz.

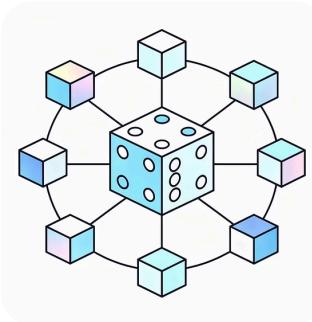
# Operações OLAP Típicas

Além de roll-up e drill-down, OLAP oferece um conjunto completo de operações que permitem manipular e visualizar cubos de dados de formas diversas, cada uma projetada para responder diferentes tipos de questões analíticas.



## Slice

Seleciona uma única fatia do cubo fixando uma dimensão em valor específico (ex: vendas apenas de Janeiro)



## Dice

Seleciona um subcubo definindo condições em múltiplas dimensões (ex: TVs e Monitores, Q1 e Q2, Brasil e Argentina)



## Pivot (Rotate)

Rotaciona o cubo para visualizar dados de perspectiva diferente, reorganizando dimensões nos eixos

## Roll-Up

Agregação subindo hierarquia

## Drill-Down

Detalhamento descendo hierarquia

## Slice

Seleção de fatia única

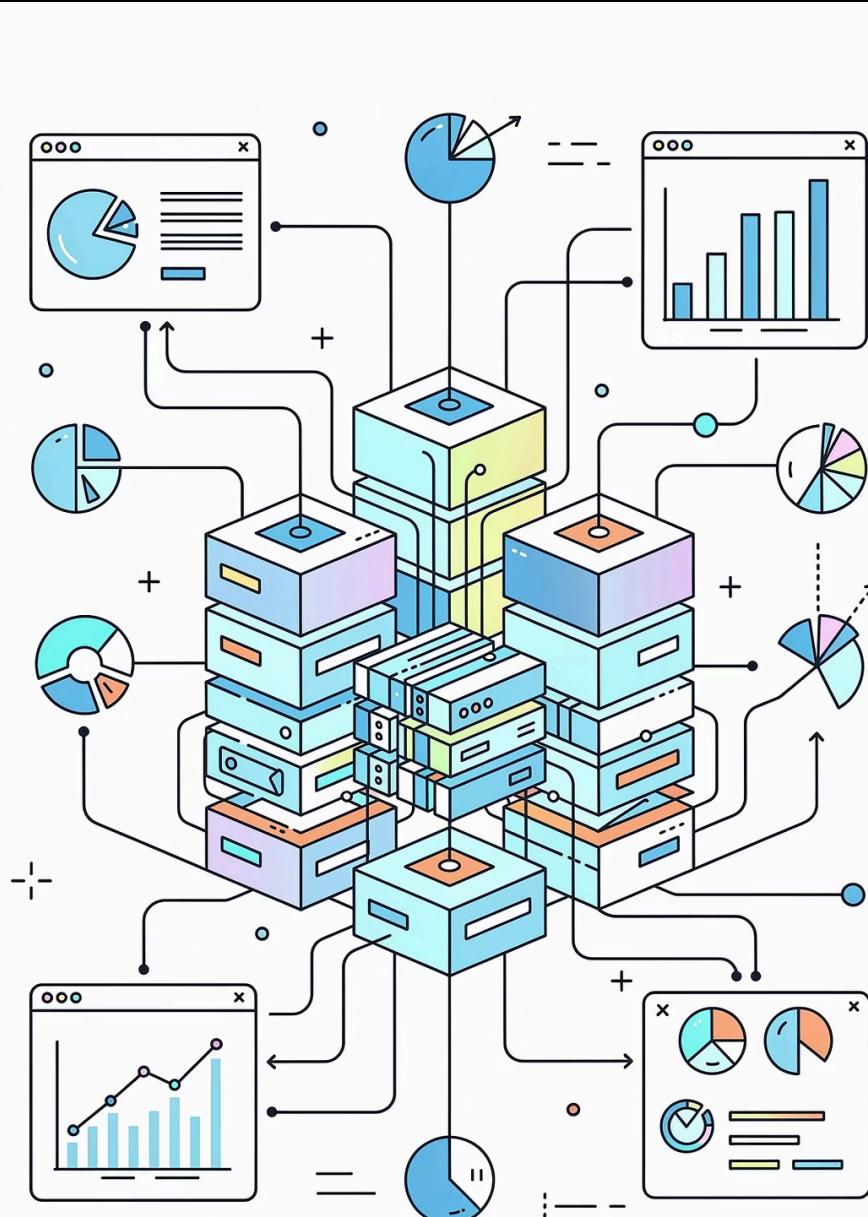
## Dice

Seleção de subcubo

## Pivot

Rotação de perspectiva

Estas operações trabalham em conjunto, permitindo que analistas naveguem fluidamente pelos dados, testando hipóteses e descobrindo insights através de exploração interativa. A combinação destas operações transforma dados brutos em inteligência de negócios acionável.



# Implementação OLAP

A implementação de sistemas OLAP (Online Analytical Processing) representa um marco fundamental na construção de soluções modernas de data warehousing. Esta apresentação explora as principais estratégias, técnicas e arquiteturas necessárias para implementar cubos OLAP eficientes e escaláveis em ambientes corporativos complexos.

## Exemplo de Consulta OLAP

### Cenário de Análise

Considere uma tabela de ensaios médicos contendo atributos como idade, gênero, sucesso do teste, entre outros. O objetivo é encontrar o número total de ensaios bem-sucedidos.

### Métodos Tradicionais:

- **Método 1:** Varredura completa da tabela uma única vez
- **Método 2:** Construção de índice B+ tree no atributo "sucesso", ainda requerendo acesso a todos os registros

### Desafio Central

É possível obter a contagem sem escanear muitos registros, ou mesmo sem acessar todos os registros de ensaios bem-sucedidos? Esta questão fundamental motiva técnicas avançadas de indexação e pré-computação em OLAP.



OTIMIZAÇÃO

## Exemplo de Join Index

### Conceito de Join Index

Um join index é uma estrutura de dados especializada que pré-computa e armazena os resultados de operações de junção entre tabelas. Isso elimina a necessidade de calcular joins repetidamente durante consultas analíticas.

### Vantagens Principais

- Redução drástica no tempo de resposta
- Minimização de I/O em disco
- Otimização para queries complexas

### Aplicação em OLAP

Em ambientes de data warehouse, join indexes são especialmente úteis para conectar tabelas fato com tabelas dimensão, acelerando consultas que envolvem múltiplas dimensões simultaneamente.

## Armazenamento Horizontal versus Vertical

A escolha da estratégia de armazenamento físico dos dados tem impacto profundo no desempenho de sistemas OLAP. Tabelas fato em data warehousing frequentemente são "gordas", contendo dezenas ou até centenas de dimensões e atributos.



### Armazenamento Horizontal

Tuplas são armazenadas sequencialmente, uma após a outra. Cada linha contém todos os atributos do registro.



### Armazenamento Vertical

Tuplas são armazenadas por atributos, com cada coluna mantida separadamente em estruturas otimizadas.

- Insight Crítico:** Consultas analíticas típicas acessam apenas alguns atributos de tabelas muito largas. O armazenamento vertical otimiza esse padrão de acesso, lendo apenas as colunas necessárias e reduzindo drasticamente o volume de dados processados.

# Armazenamento Baseado em Colunas

## Fundamentos Técnicos

O armazenamento baseado em colunas (column-based storage) revolucionou o processamento analítico ao reorganizar fisicamente os dados para otimizar consultas OLAP.

### Compressão Eficiente

Valores similares ficam agrupados, permitindo taxas de compressão superiores

### I/O Reduzido

Apenas colunas relevantes são lidas do disco

### Processamento Vetorizado

Operações são executadas em blocos de dados, aproveitando cache de CPU

## Benefícios em OLAP

- Aceleração de consultas agregadas (SUM, AVG, COUNT)
- Menor consumo de memória durante processamento
- Melhor desempenho em queries seletivas
- Escalabilidade para tabelas muito largas

Sistemas modernos como Amazon Redshift, Google BigQuery e Apache Parquet utilizam esta abordagem como fundamento arquitetural.

# Data Cube: Uma Lattice de Cuboids

## Conceitos Hierárquicos

### Base vs. Células Agregadas

Células base contêm dados granulares; células agregadas armazenam sumarizações pré-computadas

### Ancestrais vs. Descendentes

Relações de agregação entre níveis diferentes da hierarquia dimensional

### Pais vs. Filhos

Conexões diretas entre níveis adjacentes no lattice

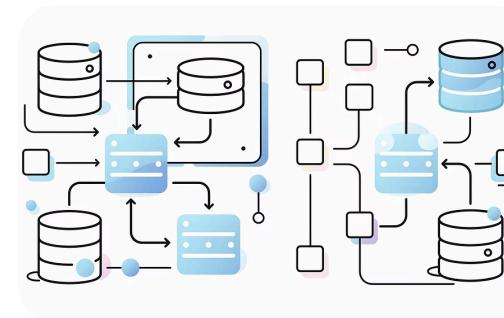
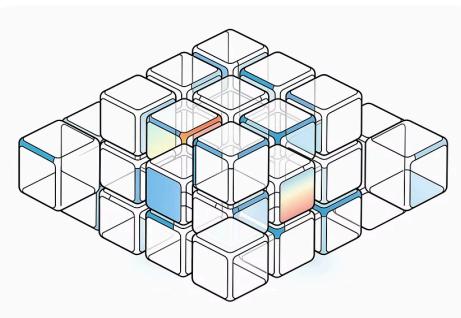
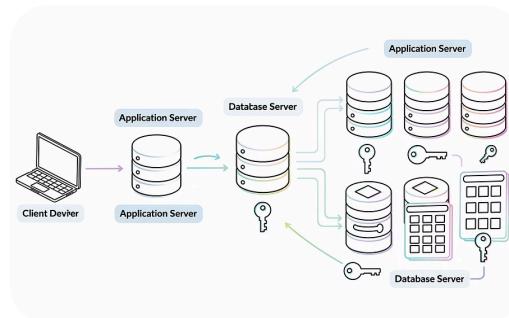
## Exemplo de Navegação

Considere um cubo de vendas com dimensões: data, produto, localização e loja.

- $(*, *, *)$  — Agregação total (topo do lattice)
- $(*, \text{milk}, *, *)$  — Todas as vendas de leite
- $(*, \text{milk}, \text{Urbana}, *)$  — Leite em Urbana
- $(*, \text{milk}, \text{Chicago}, *)$  — Leite em Chicago
- $(9/15, \text{milk}, \text{Urbana}, *)$  — Leite em Urbana no dia 9/15
- $(9/15, \text{milk}, \text{Urbana}, \text{Dairy\_land})$  — Célula base mais granular

# Arquiteturas de Servidores OLAP

A escolha da arquitetura de servidor OLAP determina o equilíbrio entre desempenho, escalabilidade e flexibilidade. Cada abordagem oferece trade-offs específicos para diferentes casos de uso corporativos.



## ROLAP (Relational OLAP)

Utiliza SGBD relacional ou relacional estendido para armazenar e gerenciar dados do warehouse, com middleware OLAP intermediário. Inclui otimização de backend, lógica de navegação de agregações e ferramentas adicionais.

**Vantagem:** Maior escalabilidade para grandes volumes.

## MOLAP (Multidimensional OLAP)

Motor de armazenamento multidimensional baseado em arrays esparsos. Oferece indexação rápida para dados sumarizados pré-computados.

**Vantagem:** Desempenho superior em consultas complexas.

## HOLAP (Hybrid OLAP)

Combina flexibilidade de abordagens relacionais e multidimensionais. Exemplo: Microsoft SQL Server. Baixo nível usa modelo relacional; alto nível emprega estruturas de array.

**Vantagem:** Equilíbrio entre desempenho e escalabilidade.

## Servidores SQL Especializados

Exemplos incluem Redbricks e outros sistemas otimizados. Suporte especializado para consultas SQL sobre esquemas star/snowflake. **Vantagem:** Otimizações específicas para workloads analíticos.

# Materialização de Cubos: Full Cube vs. Iceberg Cube

## Comparação de Abordagens

A materialização completa de cubos pode ser computacionalmente proibitiva. A abordagem Iceberg Cube oferece uma alternativa pragmática ao computar apenas células que satisfazem condições específicas.

## Sintaxe Iceberg Cube

```
COMPUTE CUBE sales ICEBERG AS  
SELECT month, city, customer_group, COUNT(*)  
FROM salesInfo  
CUBE BY month, city, customer_group  
HAVING COUNT(*) >= min_support
```

- ❑ **Condição Iceberg:** Computa apenas células cuja medida satisfaz o critério definido (ex: COUNT(\*) >= 100).

## Vantagens do Iceberg

### 1 Eficiência de Espaço

Apenas uma pequena porção de células pode estar "acima da água" em um cubo esparso

### 2 Redução de Computação

Evita processar células com valores insignificantes

### 3 Focado em Insights

Concentra recursos em padrões estatisticamente relevantes

# OLAP de Alta Dimensionalidade: A Maldição da Dimensionalidade

## Cenários de Alta Dimensionalidade

OLAP de alta dimensionalidade é necessário em múltiplas aplicações críticas:

- **Análise científica e de engenharia:** Simulações complexas com centenas de variáveis
- **Bio-data analysis:** Milhares de genes e marcadores biológicos
- **Pesquisas estatísticas:** Centenas de variáveis demográficas e comportamentais

## Exemplo Ilustrativo

Base de dados com 600.000 tuplas. Cada dimensão tem cardinalidade de 100 e distribuição Zipf de 2. O espaço de possibilidades explode exponencialmente.

## Limitações dos Métodos Tradicionais



### Iceberg e Cubos Comprimidos

Apenas atrasam a explosão combinatória inevitável

### Materialização Completa

Overhead significativo no acesso a resultados em disco

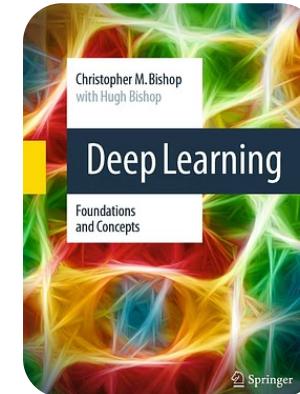
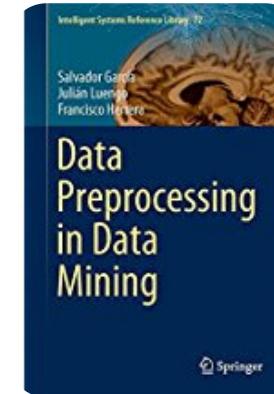
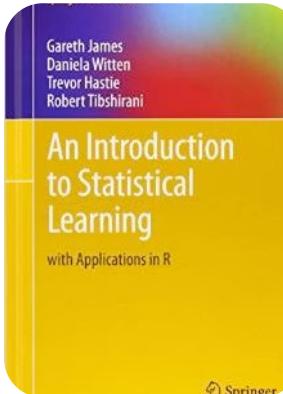
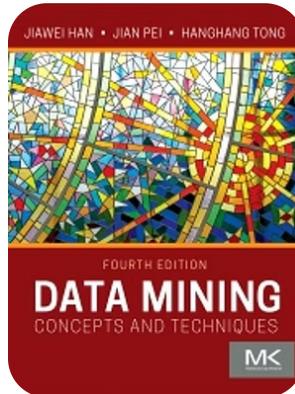
### Abordagem Shell-Fragment

Solução proposta por X. Li, J. Han, e H. Gonzalez (VLDB'04): técnica de cubing mínima para alta dimensionalidade

- ❑ **Conclusão:** Nenhum método anterior de cubing consegue lidar adequadamente com alta dimensionalidade. Novas abordagens algorítmicas são essenciais para viabilizar OLAP em cenários complexos modernos.

## Referências Principais

Esta seleção de referências representa os pilares fundamentais para o estudo aprofundado de mineração de dados, cobrindo desde conceitos básicos até técnicas avançadas e aplicações contemporâneas.



1. **J. Han, J. Pei, and H. Tong**, *Data Mining: Concepts and Techniques*, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022.
2. **G. M. James, D. Witten, T. Hastie, and R. Tibshirani**, *An Introduction to Statistical Learning: With Applications in R*. Springer Nature, 2021.
3. **S. Garcia, J. Luengo, and F. Herrera**, *Data Preprocessing in Data Mining*. Springer, 2014.
4. **C. M. Bishop and H. Bishop**, *Deep Learning: Foundations and Concepts*. Springer Nature, 2023.