Projetando um Data Warehouse

PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul

– Aula 16 –
Coleta, Preparação e Análise de Dados

Prof. Me. Lucas R. C. Pessutto

Slides adaptados do material do Prof. Lucas Silveira Kupssinskü e do Prof. Luan Fonseca Garcia

Agenda

- Data Warehouse
- Modelagem de Data Warehouse: schema e medidas
- Operações OLAP
- Cálculo de Data cubes

Definição de SAD

- Sistema de Apoio à Decisão (SAD)
  - Sistema de informação que dá suporte a atividades de tomada de decisão.
  - Objetivo: transformar dados em informação útil à tomada de decisão por parte dos interessados *(stakeholders)*

  - Exemplos:
    - Determinar mercado-alvo de produto
    - Definir preço de um produto e criar promoções e condições de compra
    - Verificar eficácia de campanhas de marketing
    - Otimizar a quantidade de produtos em estoque
    - Responder rapidamente a mudanças no mercado e determinar novas tendências

Evolução dos SAD

- Década de 60
  - '*Master Files*'
  - Fitas magnéticas


Mídia barata


Armazenamento de "grandes volumes de dados"


Acesso sequêncial!


Para acessar *n*-ésimo registro, necessário acessar (*n-1*) registros anteriores


Tempo médio de acesso a uma fita inteira: 20 a 30 minutos!!

Evolução dos SAD

- Metade da década de 60
  - Explosão de ‘*master files*’ e fitas magnéticas
  - MUITO dado redundante
    - Sincronização de arquivos torna-se necessária
    - Alta complexidade de manutenção de programas
    - Necessidade de MUITO HARDWARE para suporte a tantos arquivos!

Evolução dos SAD

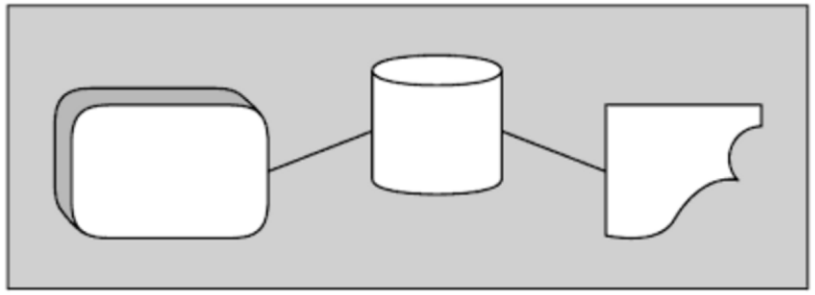
- Metade da década de 60
  - Explosão de '*master files*' e fitas magnéticas
  - MUITO dado redundante
    - Sincronização de arquivos torna-se necessária
    - Alta complexidade de manutenção de programas
    - Necessidade de MUITO HARDWARE para suporte a tantos arquivos!
  - E se a única mídia de armazenamento ainda fosse até hoje a fita magnética?
    - Jamais teríamos sistemas transacionais automatizados!
      - Sistemas de reservas, banking, etc. etc. etc.

Evolução dos SAD

- Década de 70
  - Advento do DASD (*direct access storage device* - dispositivo de armazenamento de acesso direto)
    - Basta saber o endereço do registro para acessá-lo
    - Tempo de acesso na ordem de milissegundos
  - Com DASD vieram os SGBDs
    - Armazenamento e recuperação de dados em DASD
    - Indexação de dados
    - …
  - Metade da década de 70 surge ferramental OLTP
    - *On-line transaction processing*
    - Controle de transações torna possível a existência de novos modelos de negócio e sistemas de informação
      - Controle de estoque, caixas automáticos de bancos, reservas em hotéis, etc.

Evolução dos SAD

- Década de 80
  - PCs + 4GL + Modelagem Relacional (SQL)
  - Dá para fazer mais do que processar transações!
    - Análise de dados
    - Sistemas voltados à gerência
    - Apoio à decisão!
  - Problema: paradigma de base de dados única
    - Como utilizar uma única base de dados para controle operacional (transacional) e processamento analítico?



The single-database-serving-all-purposes paradigm

Building the Data Warehouse, 4ª ed., W.H. Inmon

Evolução dos SAD

Solução (da época! não ideal!): programas de extração

1. Programa que filtra (seleciona) dados e os transfere para outra base de dados;
2. Evita problemas de desempenho quando muitos dados precisam ser analisados;
3. Gerentes passam a ter controle sobre os dados extraídos

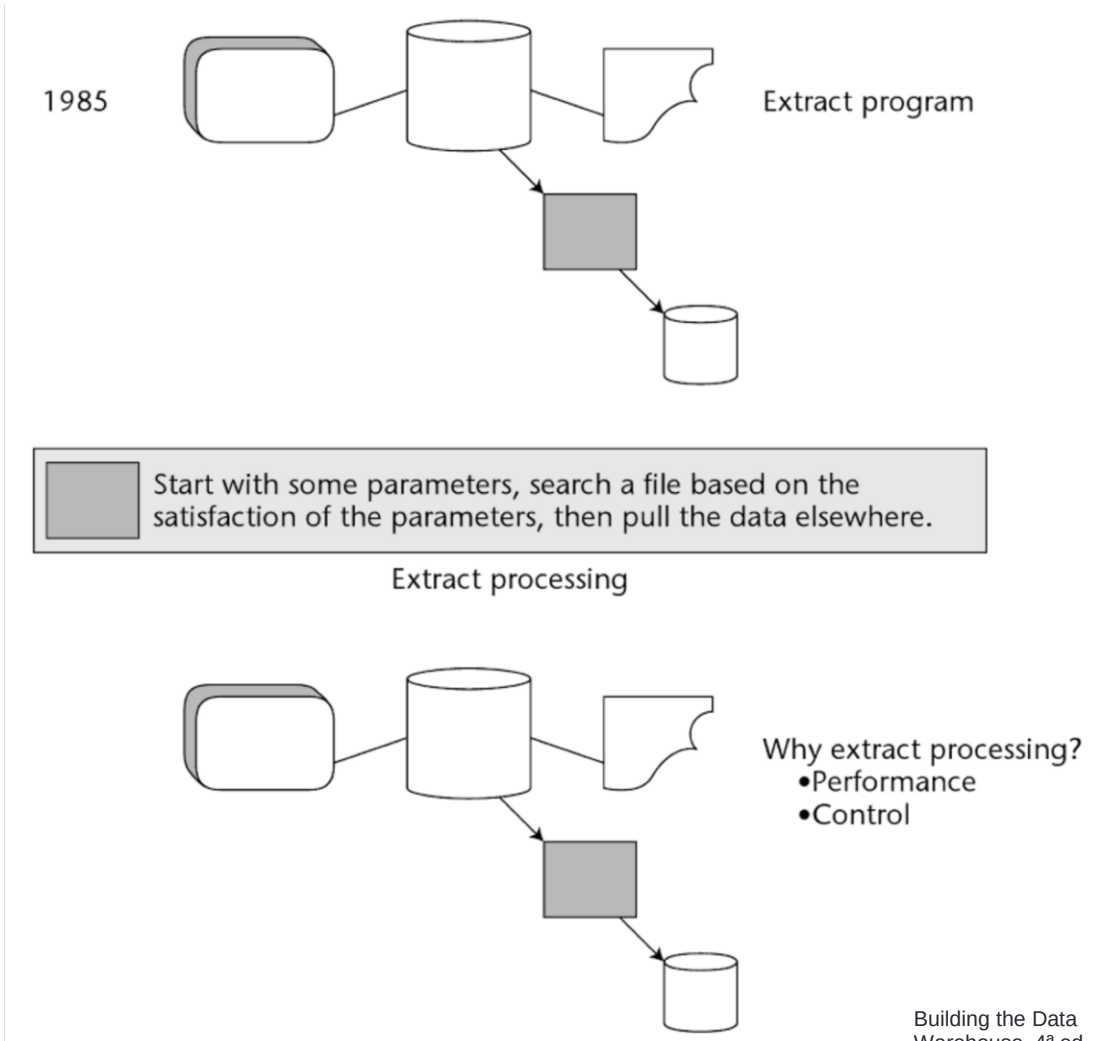
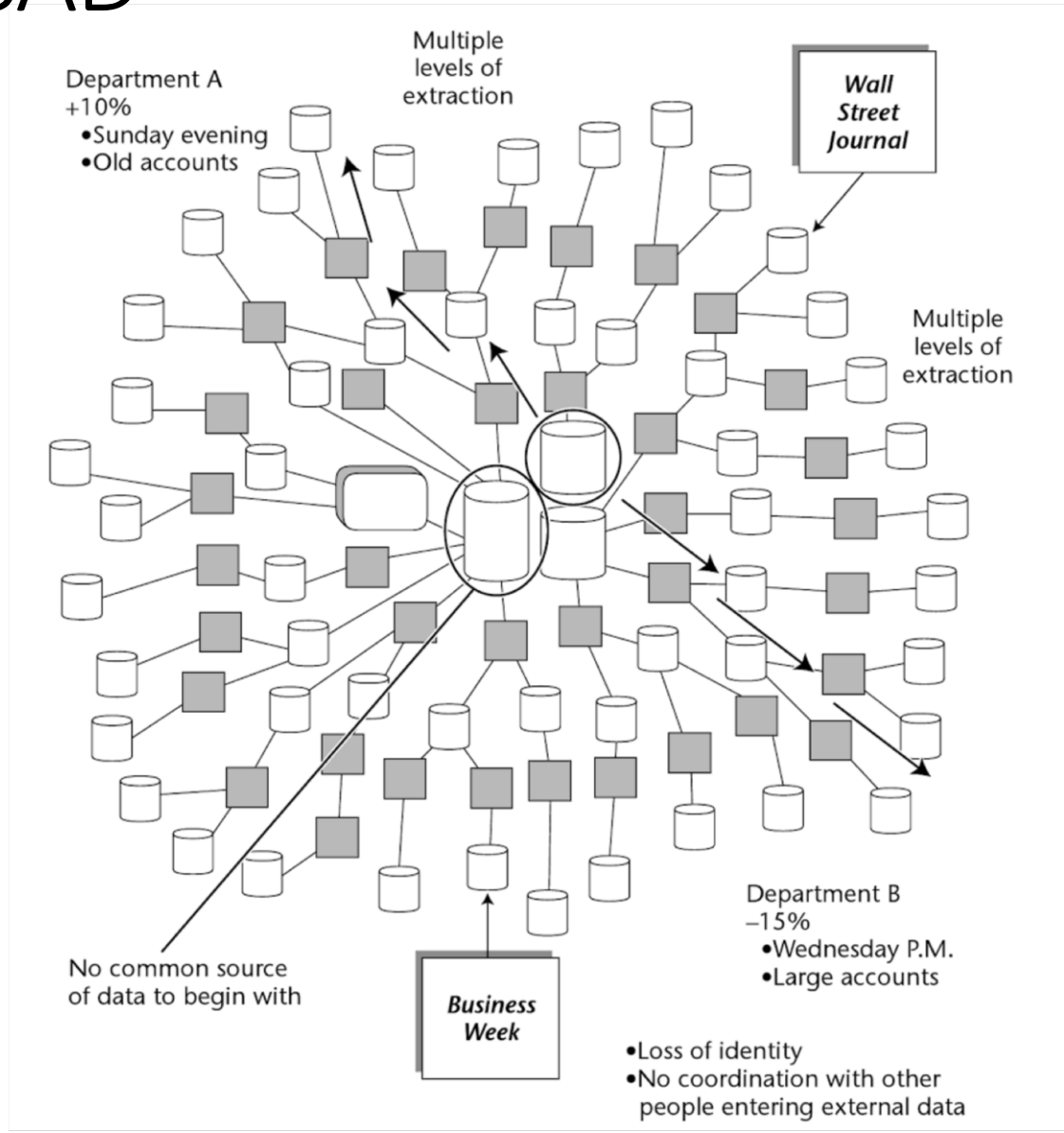


Figure 1-2 The nature of extract processing.

Building the Data Warehouse, 4ª ed., W.H. Inmon

Evolução dos SAD

- Programas de extração ficaram fora de controle
  - Extração de dados da base principal
  - Extração da extração
  - Extração da extração da extração
  - …
  - Grande empresa chegava a fazer <u>45 mil extrações/dia</u>
  - *“Arquitetura que evoluía naturalmente”*

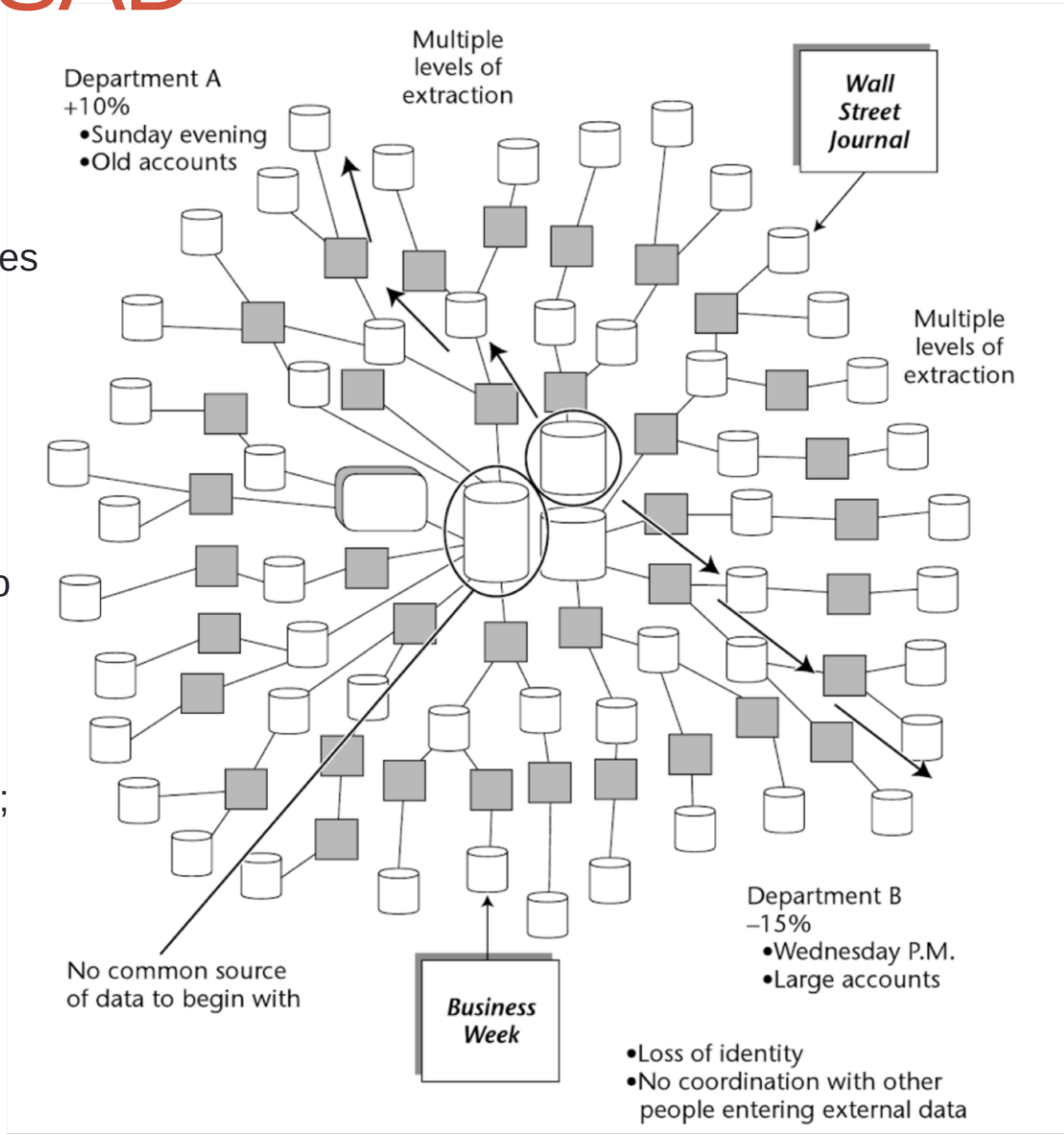


Building the Data Warehouse, 4ª ed., W.H. Inmon

Evolução dos SAD

VÁRIOS problemas:

1. Não há uma perspectiva temporal coerente (extrações feitas em dias diferentes podem levar a resultados contraditórios!);
2. Diferencial algorítmico: por exemplo, comparar extrações de clientes antigos com extrações de clientes lucrativos;
3. Conflitos entre fontes externas de dados: fontes não são documentadas;
4. Não há fonte de dados comum: departamentos distintos utilizam dados de bases distintas, sem sincronização ou compartilhamento entre tais bases;
5. Não há dados históricos suficientes armazenados para tomada de decisão estratégica!

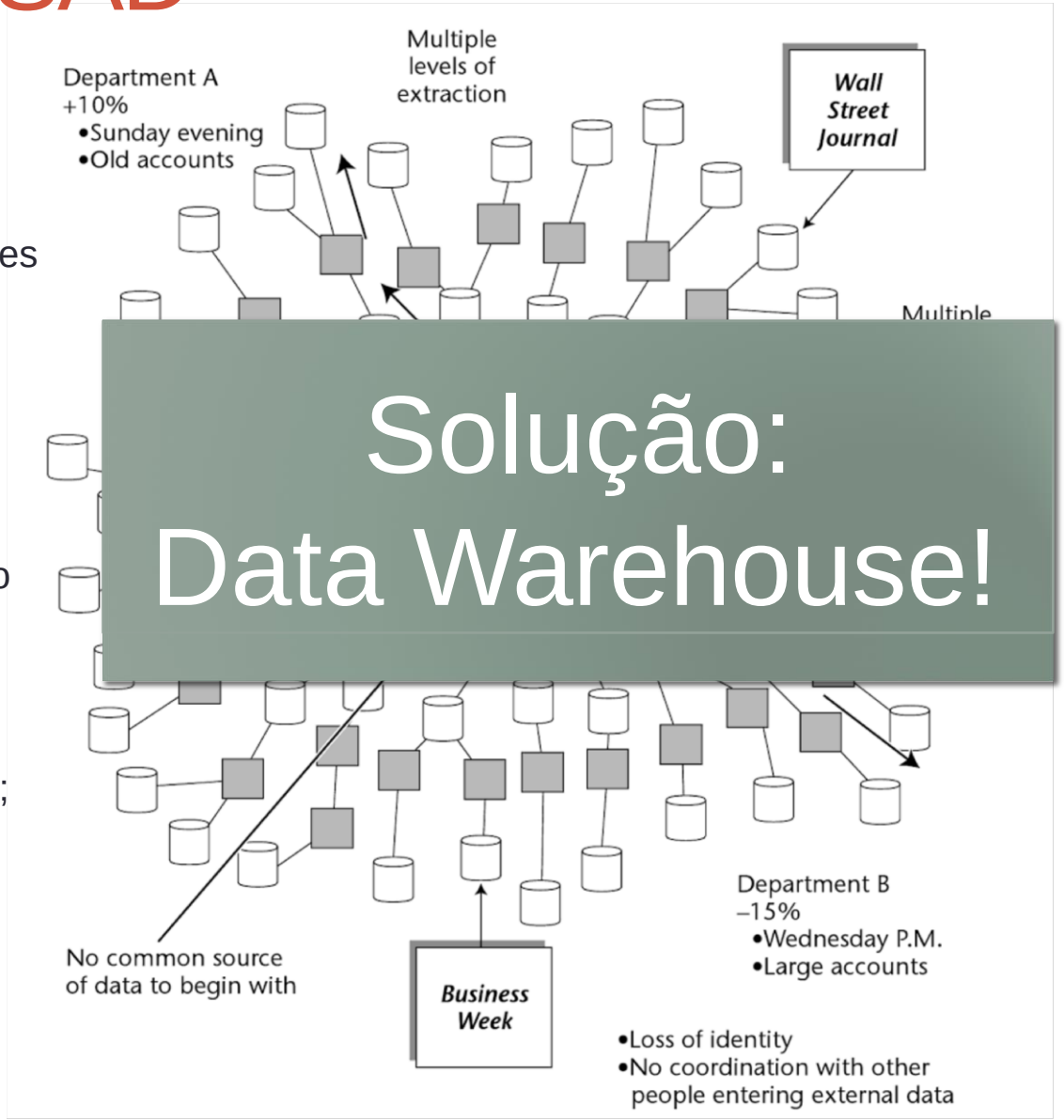


Building the Data Warehouse, 4ª ed., W.H. Inmon

Evolução dos SAD

VÁRIOS problemas:

1. Não há uma perspectiva temporal coerente (extrações feitas em dias diferentes podem levar a resultados contraditórios!);
2. Diferencial algorítmico: por exemplo, comparar extrações de clientes antigos com extrações de clientes lucrativos;
3. Conflitos entre fontes externas de dados: fontes não são documentadas;
4. Não há fonte de dados comum: departamentos distintos utilizam dados de bases distintas, sem sincronização ou compartilhamento entre tais bases;
5. Não há dados históricos suficientes armazenados para tomada de decisão estratégica!



Building the Data Warehouse, 4ª ed., W.H. Inmon

O que é Data Warehouse?

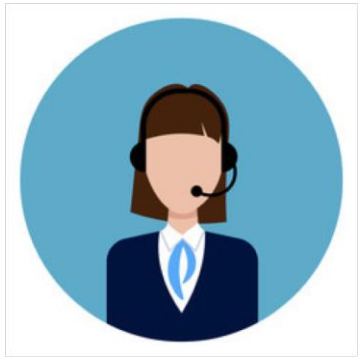
- Diversas definições:
  - Repositório de dados que é mantido de forma **separada** da base de dados operacional da organização (mas é alimentado por esses dados)
  - Suporte ao processamento de informação, fornecendo uma plataforma para análise de dados históricos consolidados

- "Um data warehouse é uma coleção de dados **orientada a assunto**, **integrada**, **temporal**, e **não volátil** para dar suporte a processos de tomada de decisão de uma organização." — W. H. Inmon

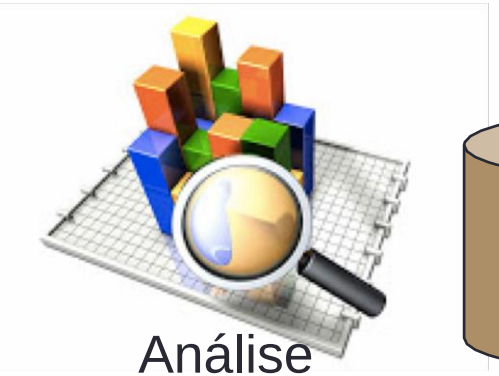
- Data warehousing:
  - O processo de construir e utilizar data warehouses

Data Warehouse – Características

- **<u>Orientação a assunto</u>**
  - Dados em um DW são organizados de modo a se facilitar a análise por parte da gestão
  - Ao contrário de bancos operacionais, onde os dados são organizados de acordo com as necessidades da aplicação, no DW os dados são organizados por assunto
  - DW provê visão simples e concisa sobre objetivos da organização, excluindo dados que não são úteis para a tomada de decisão



Aplicação
de Vendas



Análise
de Vendas

Data Warehouse – Características

- **Integração**

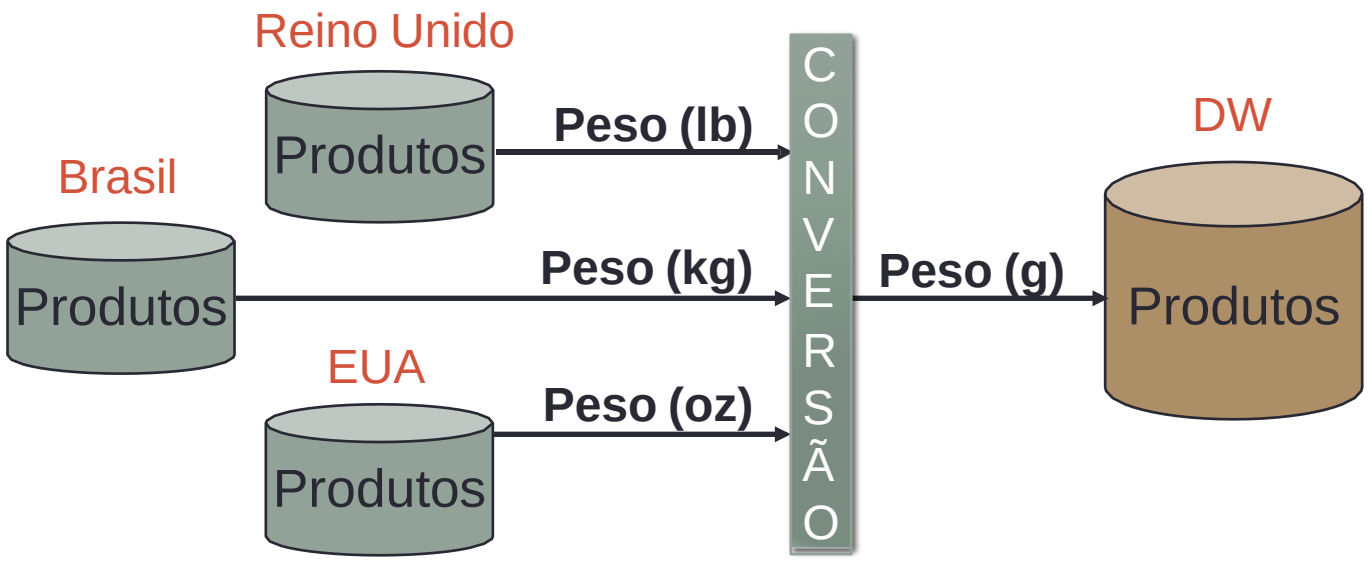
Integração de múltiplas fontes heterogêneas de dados:

Bancos relacionais, arquivos, registros de transações

Quase sempre necessário o uso de **técnicas** para **limpeza** e **integração** dos dados para garantir consistência em:

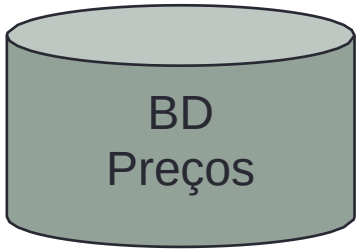
Convenção de nomes
Estruturas de representação
Medidas de atributos

Dados tipicamente são convertidos quando movidos para DW



Data Warehouse – Características

- **Variação no tempo**
  - Os dados de um DW SEMPRE possuem, implícita ou explicitamente, um componente temporal
  - BD operacional = dados correntes
  - DW = dados históricos, de longo prazo (ex: últimos 5-10 anos)



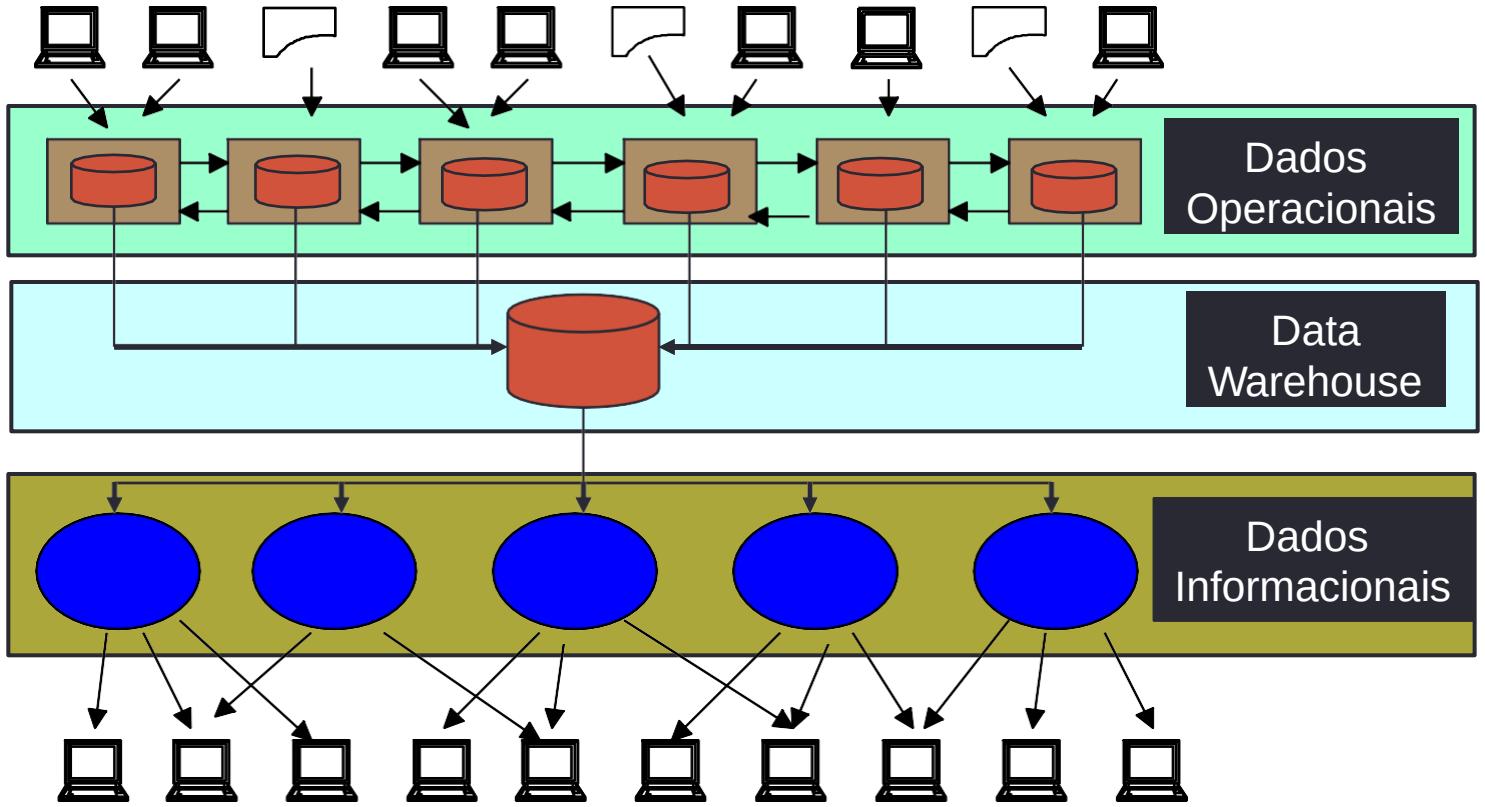
Produto	Preço
Caneta Azul	R$ 1,58
Lápis Preto	R$ 0,81
…	…

Produto	Jan/19	Fev/19	Mar/19
Caneta Azul	R$ 1,50	R$ 1,55	R$ 1,58
Lápis Preto	R$ 0,70	R$ 0,77	R$ 0,81
…	…	…	…

Data Warehouse – Características

- **Não-volatilidade**
  - BD operacional é volátil, permitindo
    - Inclusão
    - Alteração
    - Eliminação
  - DW NÃO é volátil. Operações possíveis:
    - Carga inicial
    - Consulta
  - DW dispensa:
    - Controle de transações
    - Controle de concorrência
    - Mecanismos de recuperação de dados

SAD baseado em Data Warehouse



OLTP vs. OLAP

- OLTP: Online Transactional Processing
  - Operações de SGBDs tradicionais
  - Processamento de transações e consultas
- OLAP: Online Analytical Processing
  - Operações de Data Warehouse: Drilling, slicing, dicing, etc.

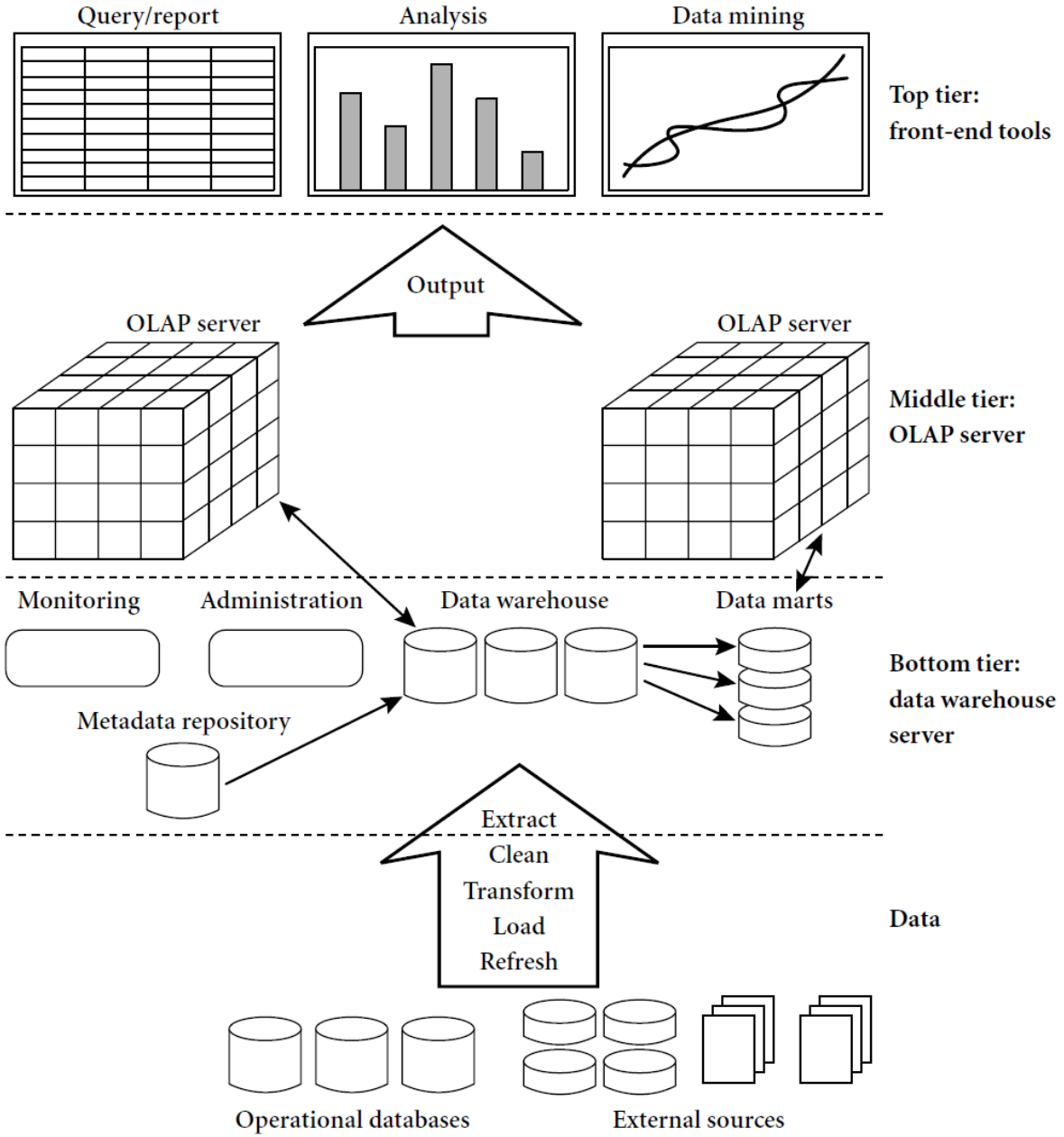
	OLTP	**OLAP**
Usuários	Atendentes, profissional de IT	Trabalhador do conhecimento
Função	Operações diárias	Suporte a decisões
Projeto do BD	Orientado a aplicações	Orientado a assunto
Dados	atuais, atualizados, detalhados, relacionais, isolados	Históricos, sumarizados, multidimensional, integrados, consolidados
Uso	Repetitivo	Sob demanda
Acesso	leitura/escrita index/hash na prim. key	Muitos “scans”
Unidade de trabalho	Transação simples e pequena	Consulta complexa
# registros acessados	dezenas	milhões
# usuários	milhares	centenas
Tamanho do BD	100MB-GB	100GB-TB

Por que Data Warehouse separado?

- Alta **performance** para ambos sistemas:
  - SGBD é ajustado para OLTP: métodos de acesso, indexação, controle de concorrência, recuperação
  - Warehouse é ajustado para OLAP: queries OLAP complexas, visão multidimensional, consolidação
- Funções e dados **diferentes**:
  - Faltam dados: Suporte a decisão requer dados **históricos**, coisa que tipicamente DBs operacionais não mantém
  - Consolidação dos dados: Dataset precisa ser **consolidado** (agregação, sumarização) de dados de fontes heterogêneas
  - Qualidade dos dados: fontes diferentes tipicamente usam representações de dados **inconsistentes**, códigos e formatos distintos que precisam ser integrados
- Cada vez mais existem sistemas que realizam análise OLAP diretamente na base relacional!

Data Warehouse: Arquitetura Multinível

- Nível mais alto:
  - Ferramentas front-end
- Nível central:
  - Servidor de OLAP
- Nível mais baixo:
  - Servidor de Data Warehouse
- Dados



Três Modelos de Data Warehouse

- **Warehouse Corporativo**
  - Mantém informações sobre assuntos ao longo de todos setores relevantes da corporação.
- **Data Mart**
  - Subconjunto de dados corporativos que possuem valor apenas pra grupos específicos dentro da empresa.
  - O escopo é limitado a grupos selecionados específicos, como por exemplo um Data Mart para Marketing
    - Data mart independente vs. dependente (direto da warehouse)
- **Warehouse Virtual**
  - Conjunto de **views** sobre bases operacionais
  - Apenas um conjunto limitado de informações pode ser sumarizado nestas views

ETL - Extraction, Transformation, and Loading

- **Data Extraction**
  - Coletar dados de múltiplas fontes heterogêneas e externas
- **Data Cleaning**
  - Identificar erros nos dados e realiza correções quando possível
- **Data Transformation**
  - Converter dados do formato original ou legado para o formato da data warehouse
- **Load**
  - Ordenar, sumarizar, consolidar, computar views, checar integridade, criar índices e partições
- **Refresh**
  - Propagar as atualizações das fontes de dados para a warehouse

Repositório de Metadados

- Metadados são dados sobre os dados que definem os objetos no warehouse:
  - Descrevem a estrutura da data warehouse
  - Metadados operacionais
    - “linhagem” dos dados (histórico de migração e transformações aplicadas), estado dos dados (ativos, arquivados, removido), informações de monitoramento (estatísticas de uso da warehouse, relatórios de erros, auditorias)
  - Algoritmos utilizados para sumarização
  - Mapeamento do ambiente de operações para o ambiente da data warehouse
  - Dados relacionados a performance do sistema

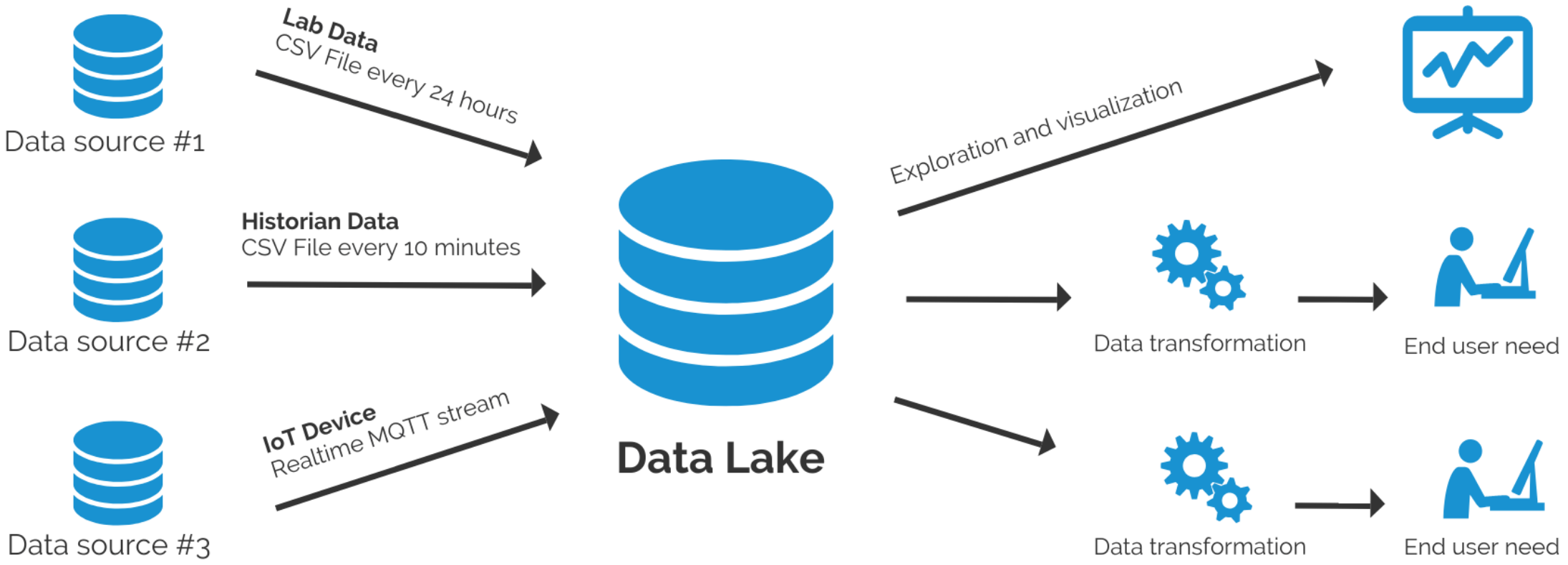
Data Lake

- É um repositório centralizado que armazena todos **dados estruturados** e **não estruturados** de qualquer escala em uma organização

- Dados são armazenados em sua forma original, sem necessariamente estruturar os dados primeiro.

- Serve como base para executar diferentes tipos de analytics para suporte a decisão:
  - Dashboards e visualizações
  - Processamento de big data
  - Analytics em tempo real
  - Machine Learning

Data Lake


Data source #1
Lab Data
CSV File every 24 hours
Data source #2
Historian Data
CSV File every 10 minutes
Data source #3
IoT Device
Realtime MQTT stream
Data Lake
Exploration and visualization
Data transformation
End user need
Data transformation
End user need

De tabelas e planilhas para Data Cubes

- Data warehouse é baseado em um modelo de dados multidimensional em que dados formam uma estrutura chamada **data cube**
- Um data cube permite que os dados sejam modelados e visualizados em múltiplas dimensões. Composto por:
  - **Tabelas de dimensões**: tabela de item (item_name, marca, tipo), tempo (dia, semana, mês, trimestre, ano)
  - **Tabelas de fatos** contendo medidas (total vendido, faturamento, etc) e chaves estrangeiras para cada um das tabelas de dimensões relacionadas

Exemplo

- Tabela 2D com valores de vendas para uma cidade
  - Valores em total de dólares vendidos

location = "Vancouver"				
time (quarter)	item (type)			
	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

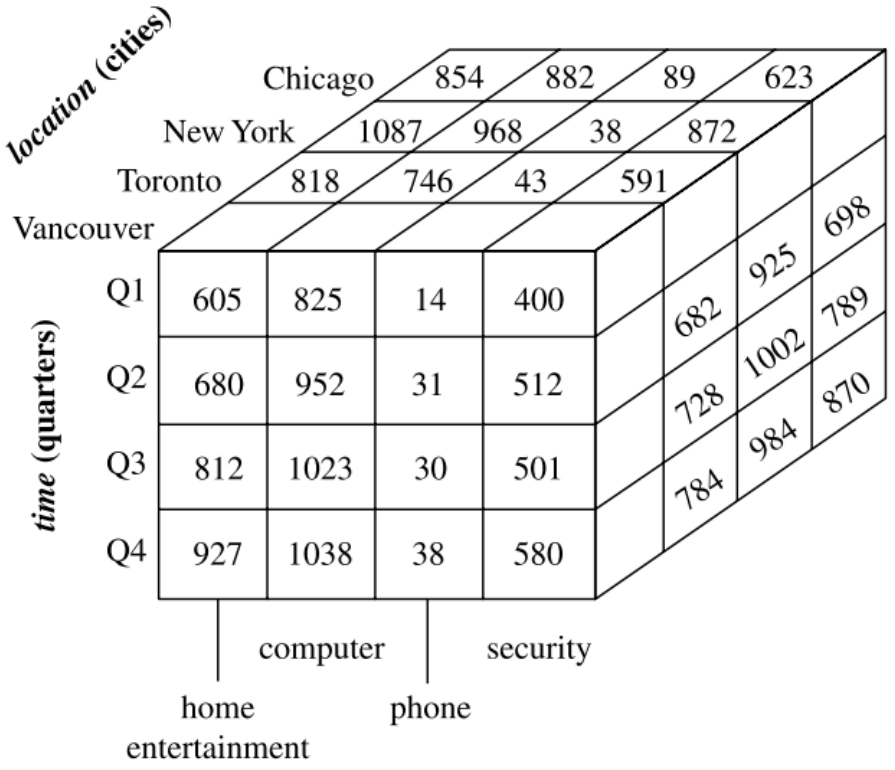
Exemplo

- Tabela 3D com valores de vendas para várias cidades
  - Valores em total de dólares vendidos

	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
time	item				item				item				item			
	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

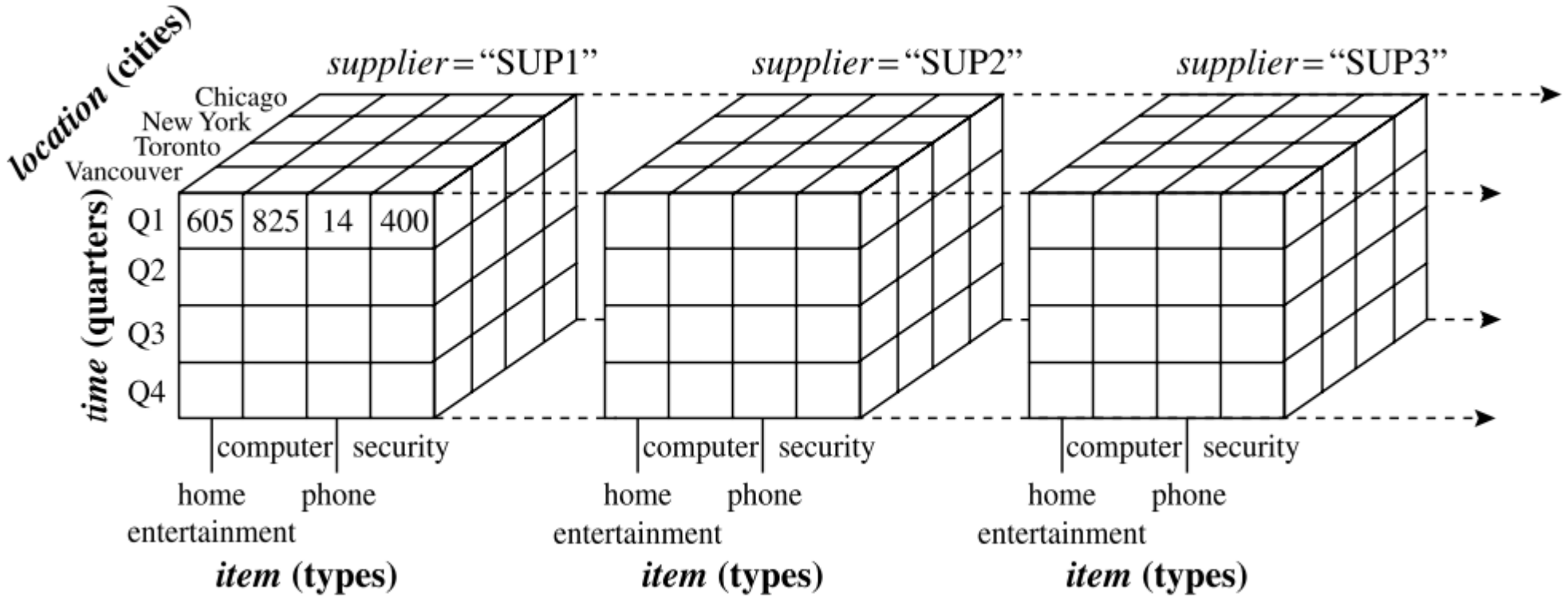
Exemplo

- Cubo 3D com valores de vendas para várias cidades



Exemplo

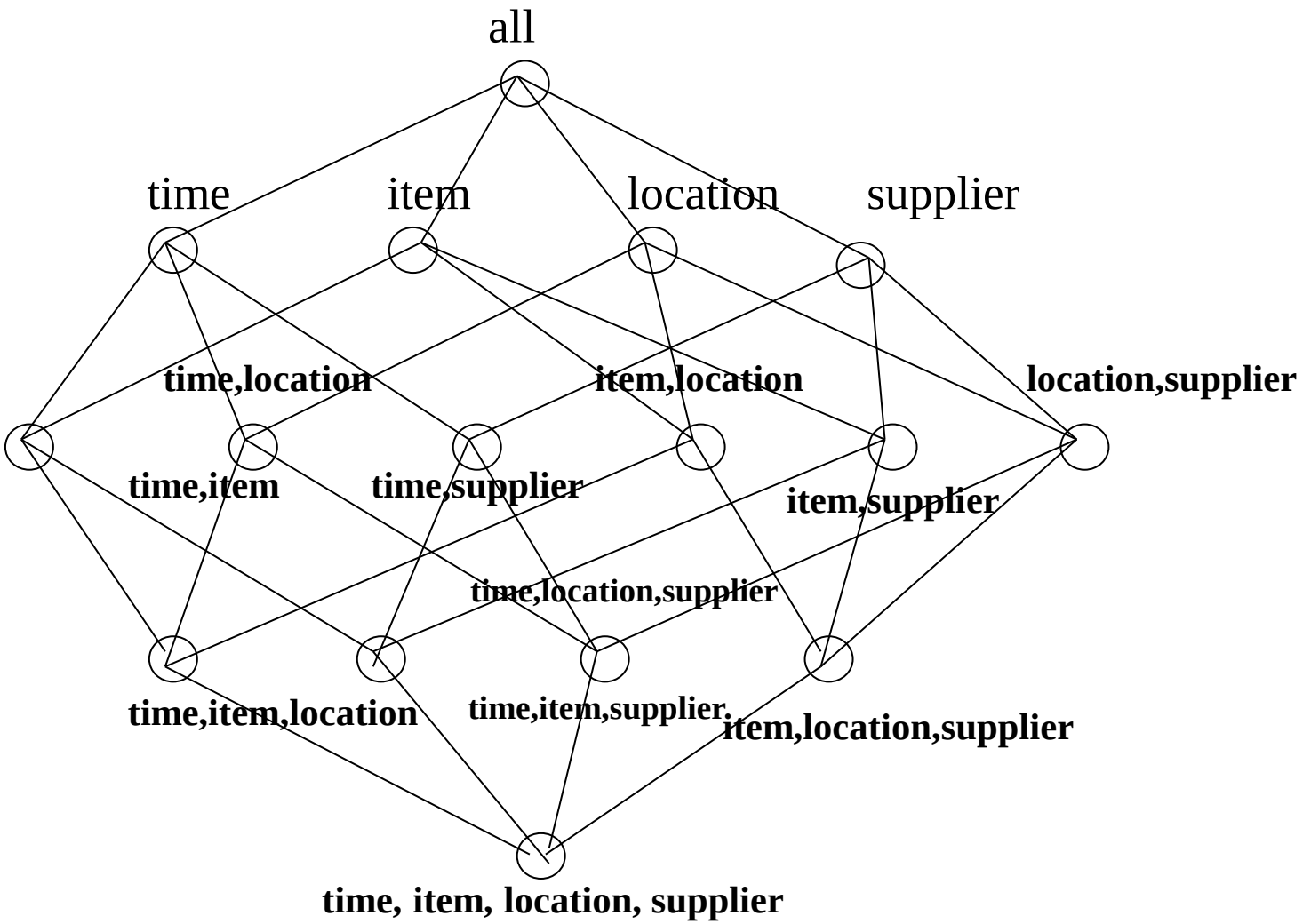
- Cubo 4D com valores de vendas por cidades e também fornecedores



Cuboids e Data Cube

- Na literatura, um cubo de n-dimensões é chamado de **cuboid**
- O cuboid que contém o nível mais baixo de sumarização é chamado de **apex cuboid**
  - Todas dimensões são agregadas!
- O cuboid que contém o nível mais alto de sumarização (nenhuma dimensão é agregada) é chamado de **cuboid base**.
  - Nenhuma dimensão é agregada!
- Um datacube é o conjunto de todos esses cuboids possíveis
- Pode ser visualizado como uma **treliça** de cuboids

Data Cube: Um reticulado de cuboids



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

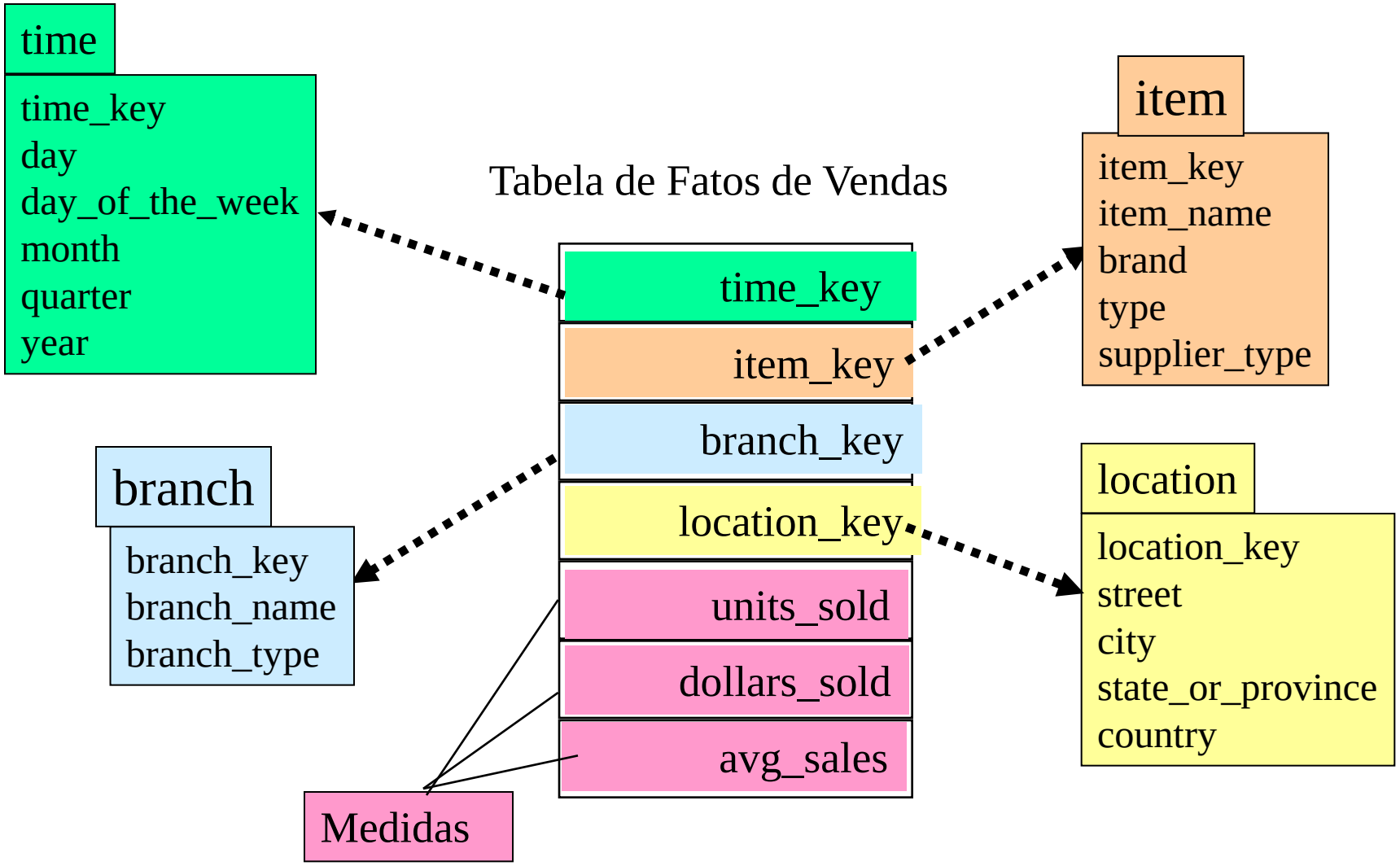
3-D cuboids

4-D (*base*) cuboid

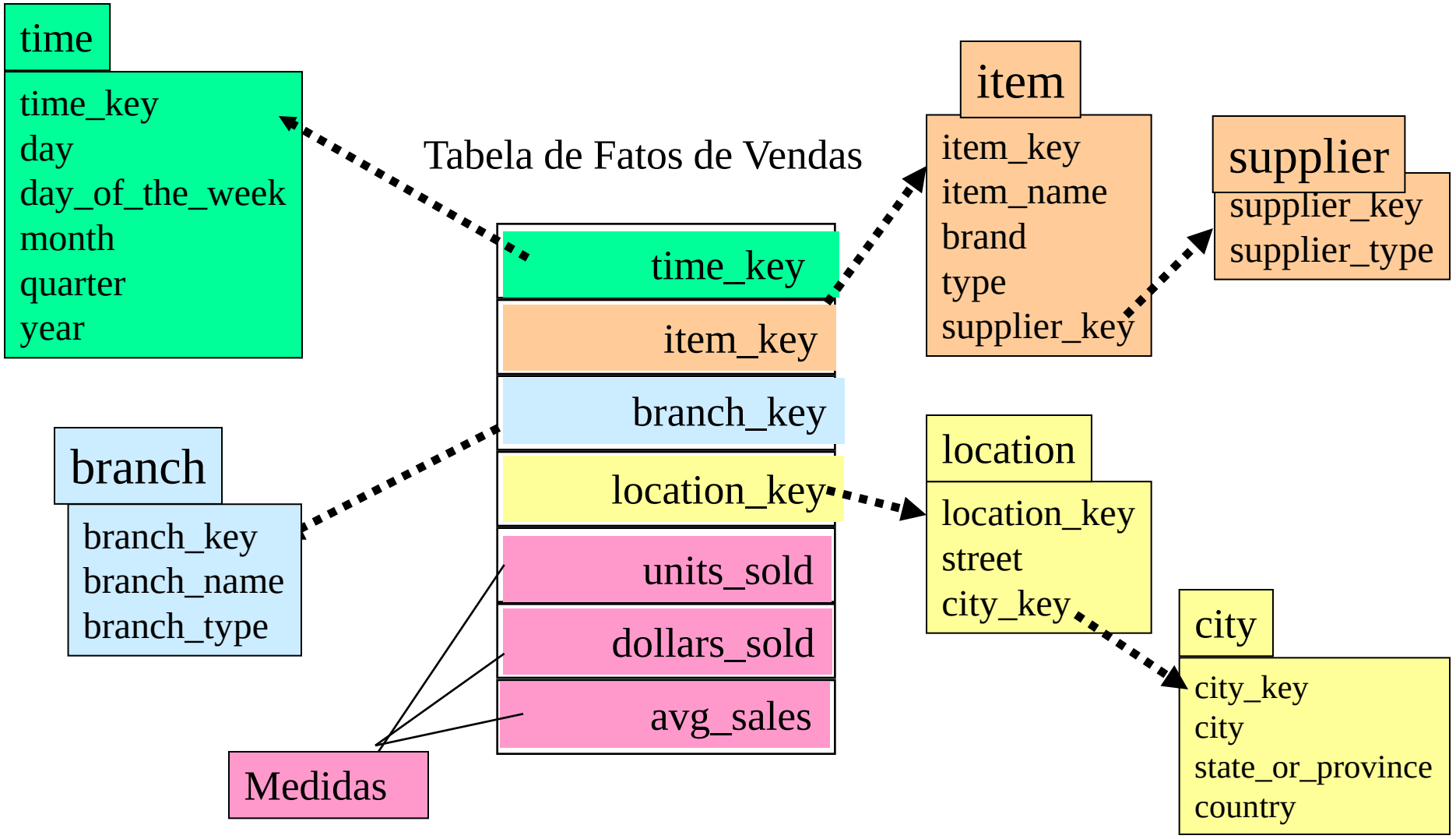
Modelagem Conceitual de Data Warehouses

- Modelando data warehouses: **dimensões** & **medidas**
- **Star schema**: uma tabela de fatos central, conectada a um conjunto de tabelas de dimensões
- **Snowflake schema**: refinamento do star schema onde alguma hierarquia dimensional é normalizada em um conjunto de tabelas de dimensões menores, criando uma forma visual que lembra um floco de neve
- **Constelações de fatos**: múltiplas tabelas de fatos que compartilham tabelas de dimensões, vista como uma coleção de star schemas, por isso chamado de schema de galáxia ou schema de constelações de fatos.

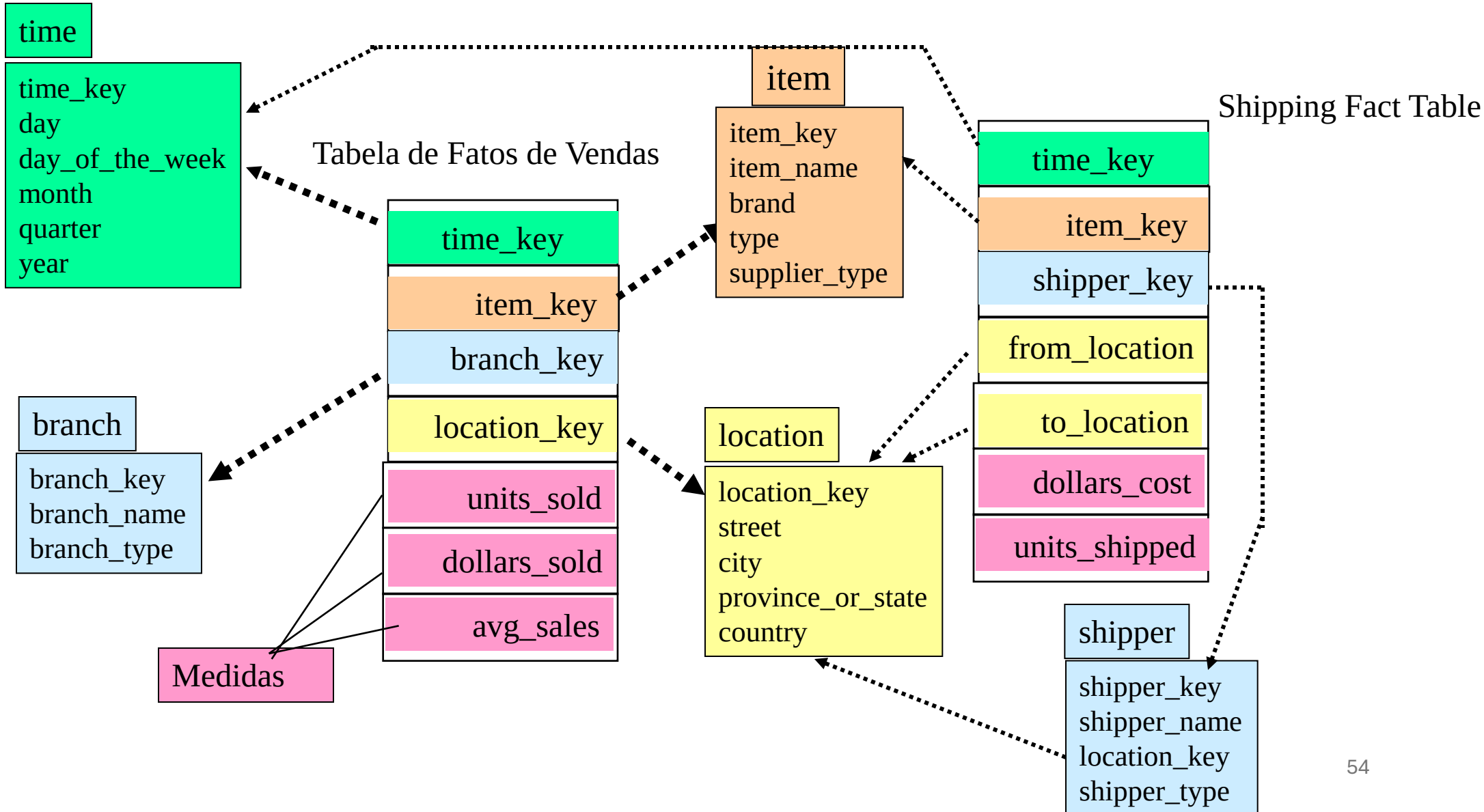
Star Schema: Exemplo



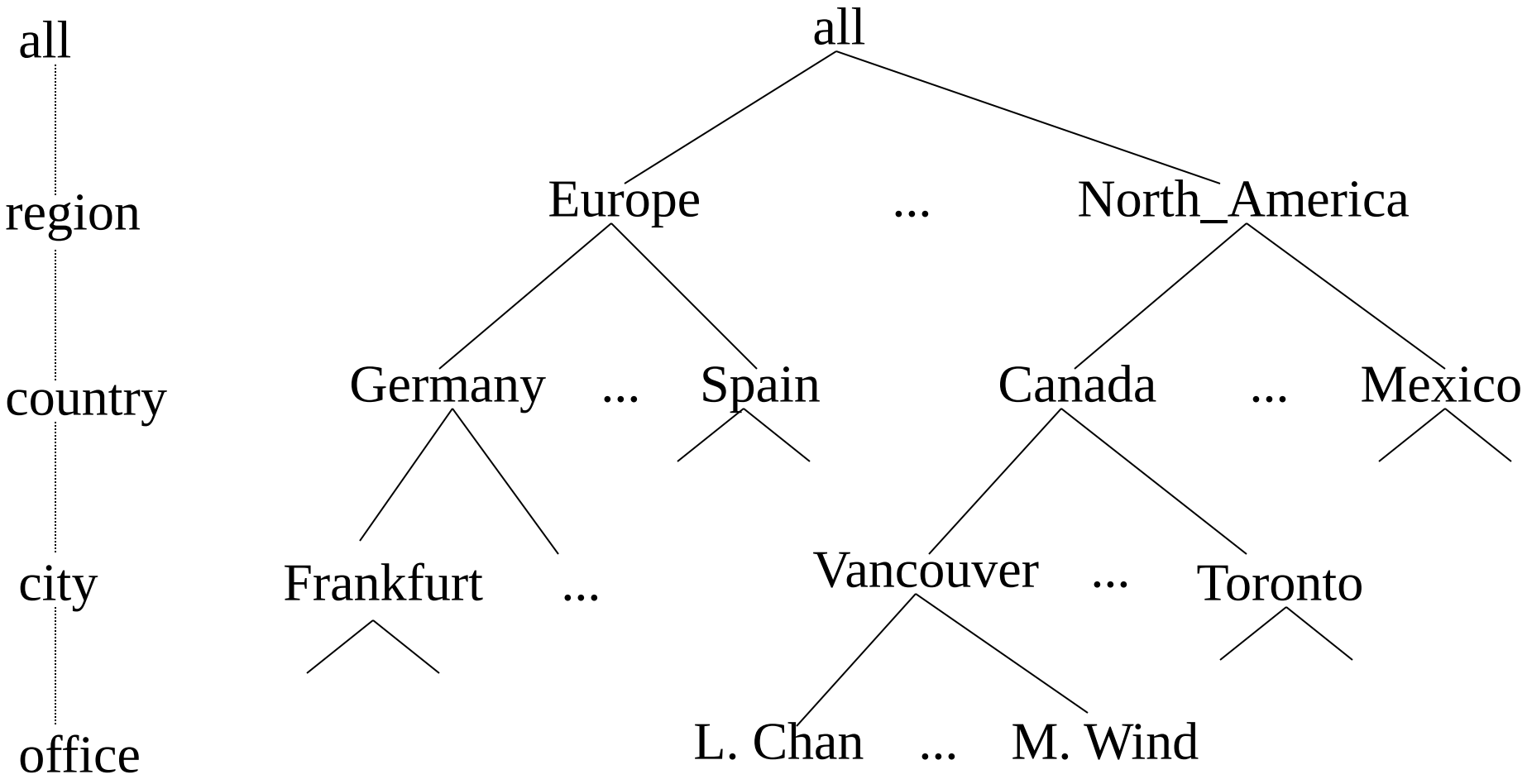
Snowflake Schema: Exemplo



Constelação de Fatos: Exemplo

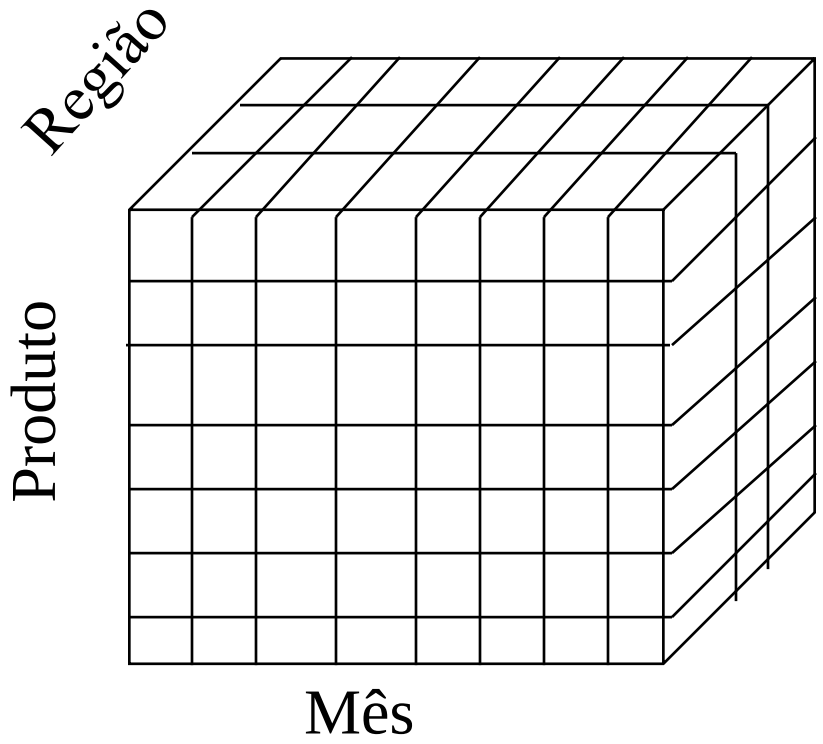


Hierarquia de Conceitos para uma Dimensão (localização)

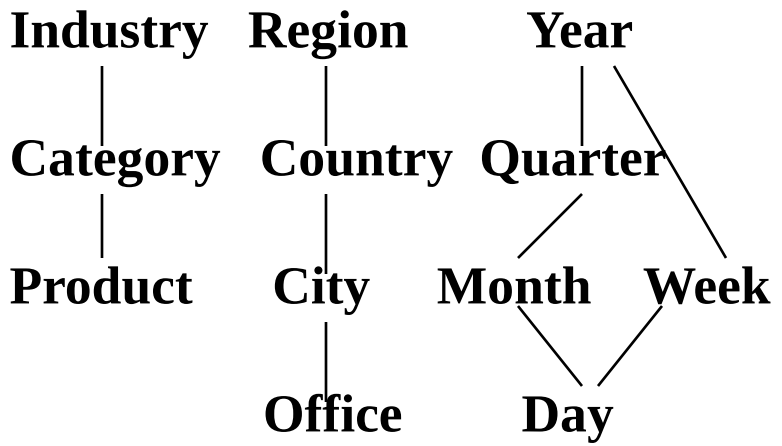


Dados Multidimensionais

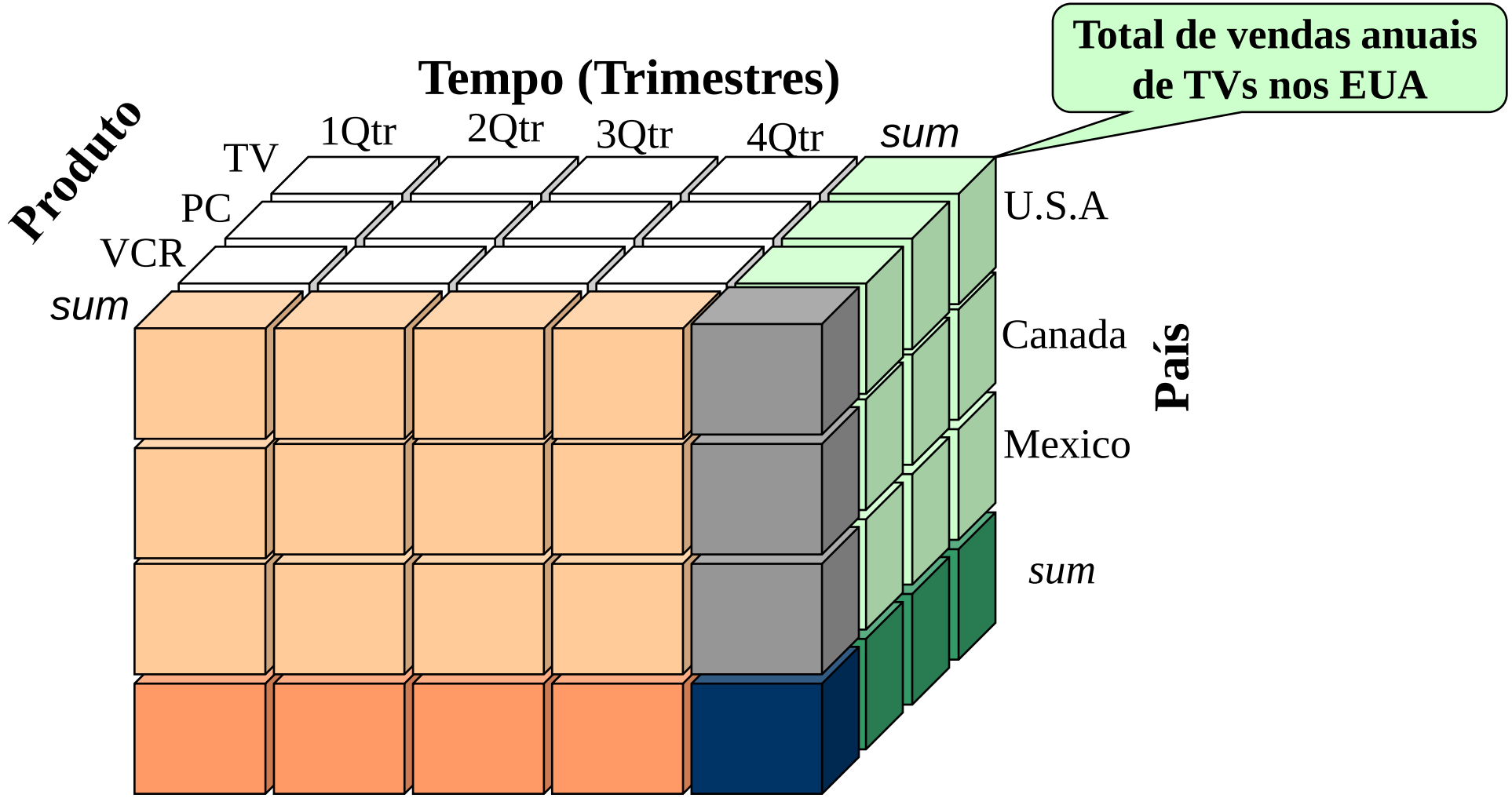
- Volume de vendas em função do produto, mês e região



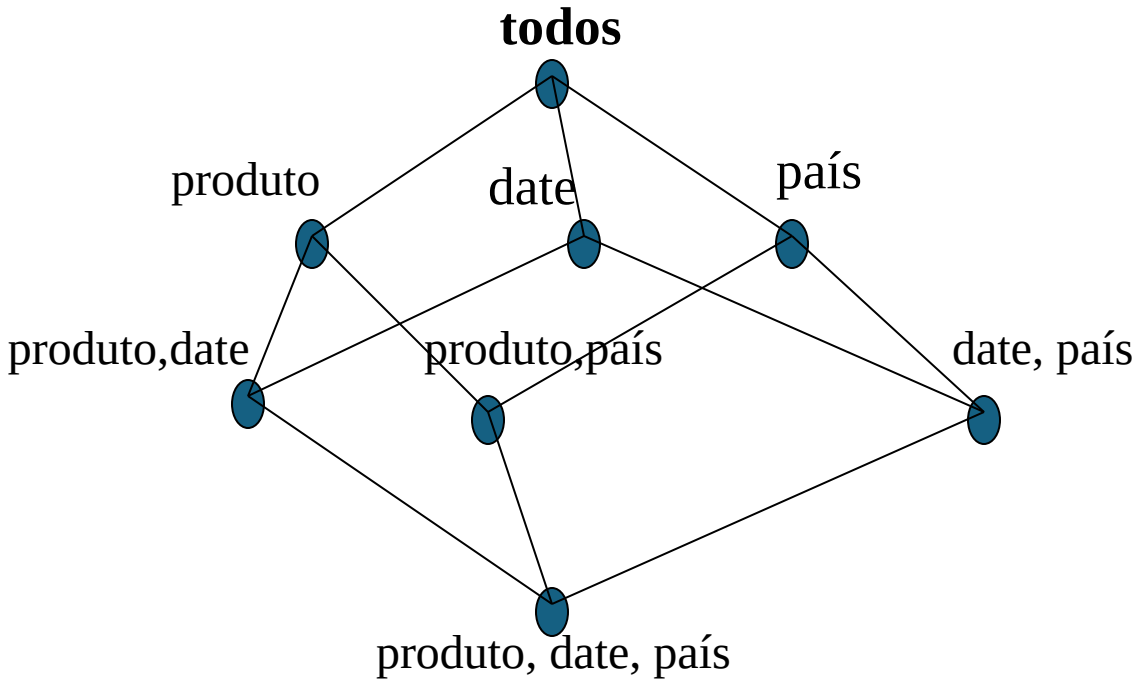
Dimensões: ***Produto, Localização, Tempo***
Sumarização Hierárquica



Exemplo de Data Cube



Cuboids que correspondem ao Cubo



0-D (*apex*) cuboid

1-D cuboids

2-D cuboids

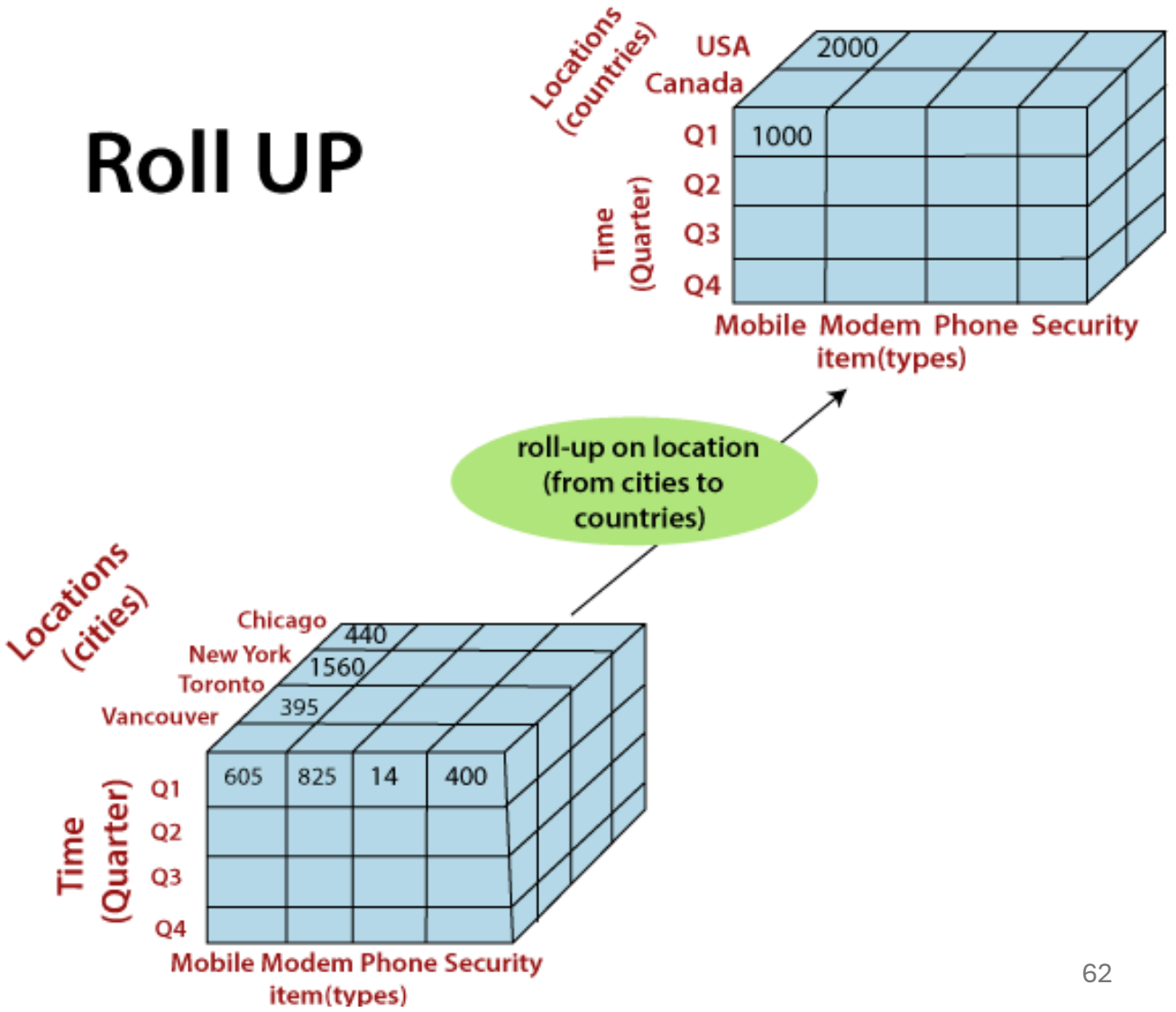
3-D (*base*) cuboid

Operações OLAP

- *Como operações OLAP multidimensionais podem ser utilizadas em análise de dados?*
- Dados estão organizados em múltiplas dimensões, onde cada dimensão contém vários níveis de abstrações de acordo com hierarquia de conceitos.
- Este tipo de organização permite flexibilidade aos usuários para criar views dos dados sob diferentes perspectivas.
- Operações OLAP são operações que fazemos em cubos para fazer consultas e análise de forma interativa com o usuário na análise de dados.

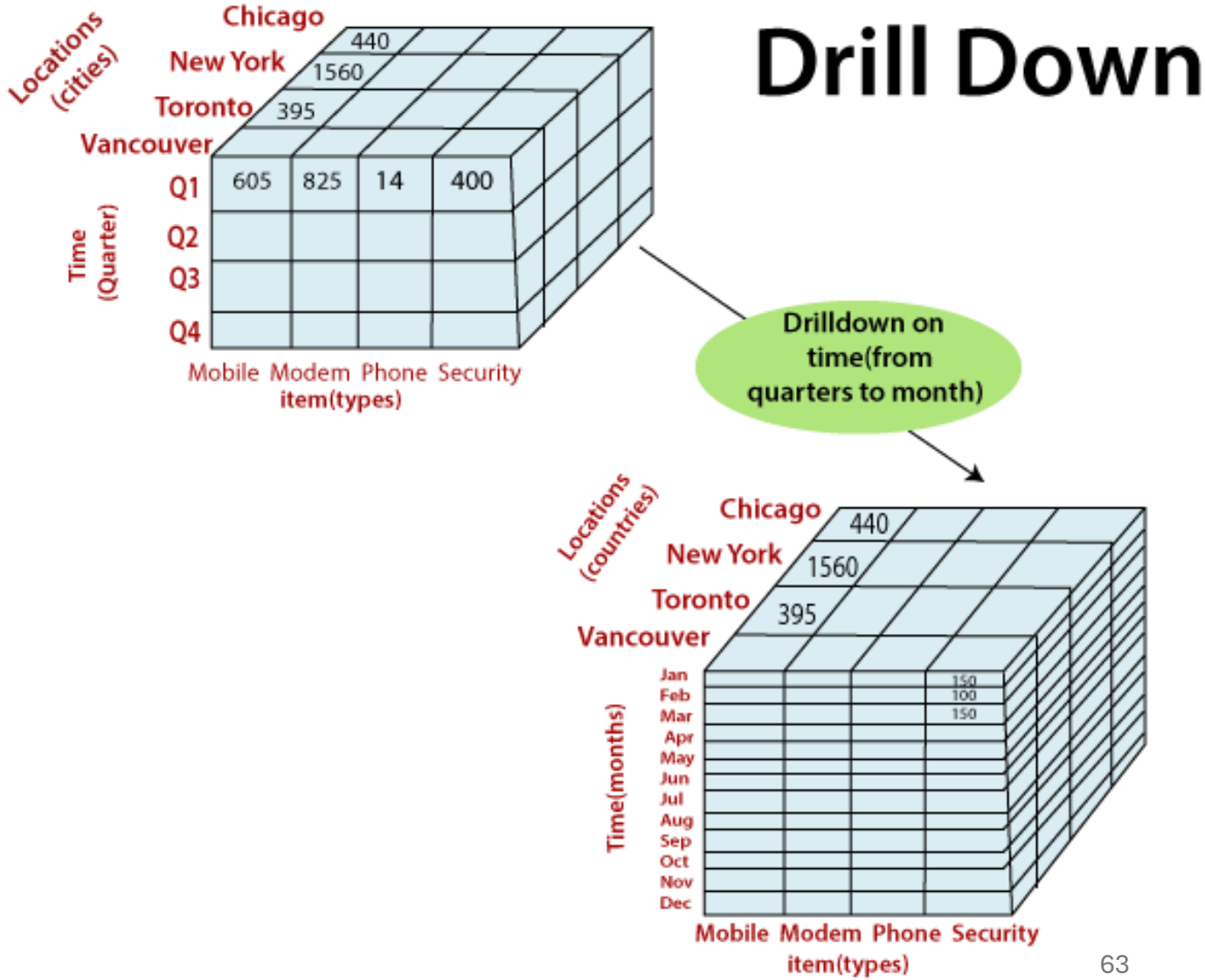
Operações OLAP – Roll Up

Roll (Drill) up: sumariza os dados subindo na **hierarquia** ou reduzindo a **dimensionalidade.**



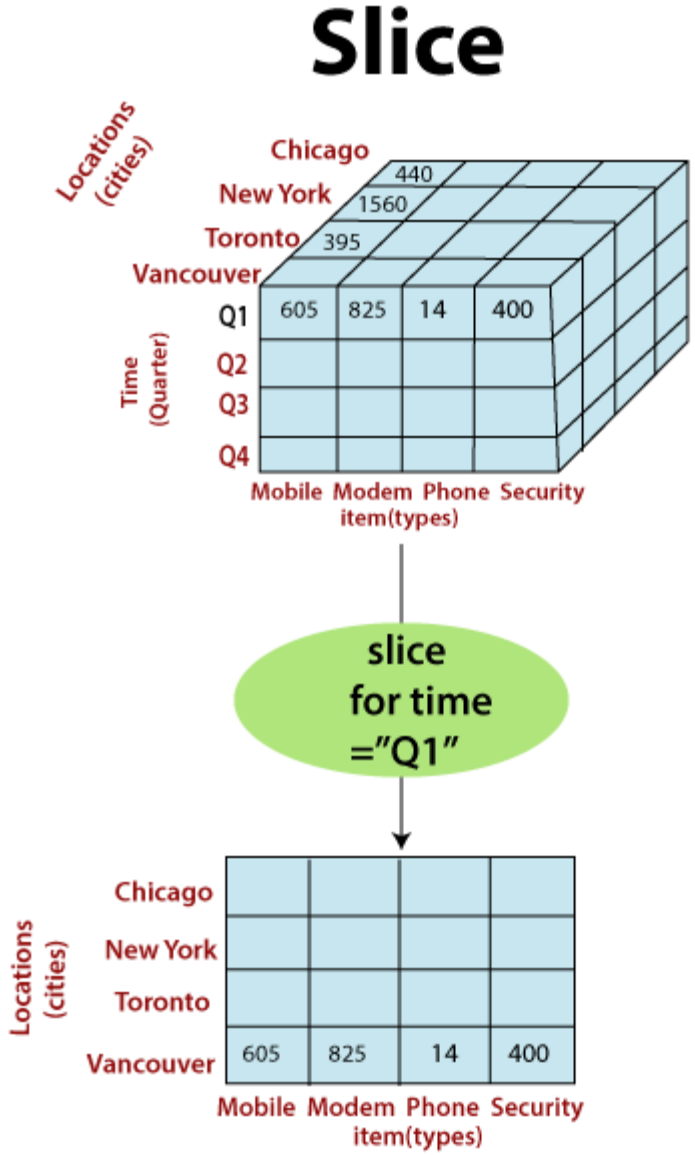
Operações OLAP – Drill Down

Drill (Roll) down: sumariza de forma inversa, **adicionando** dimensões ou **descendo** na hierarquia.



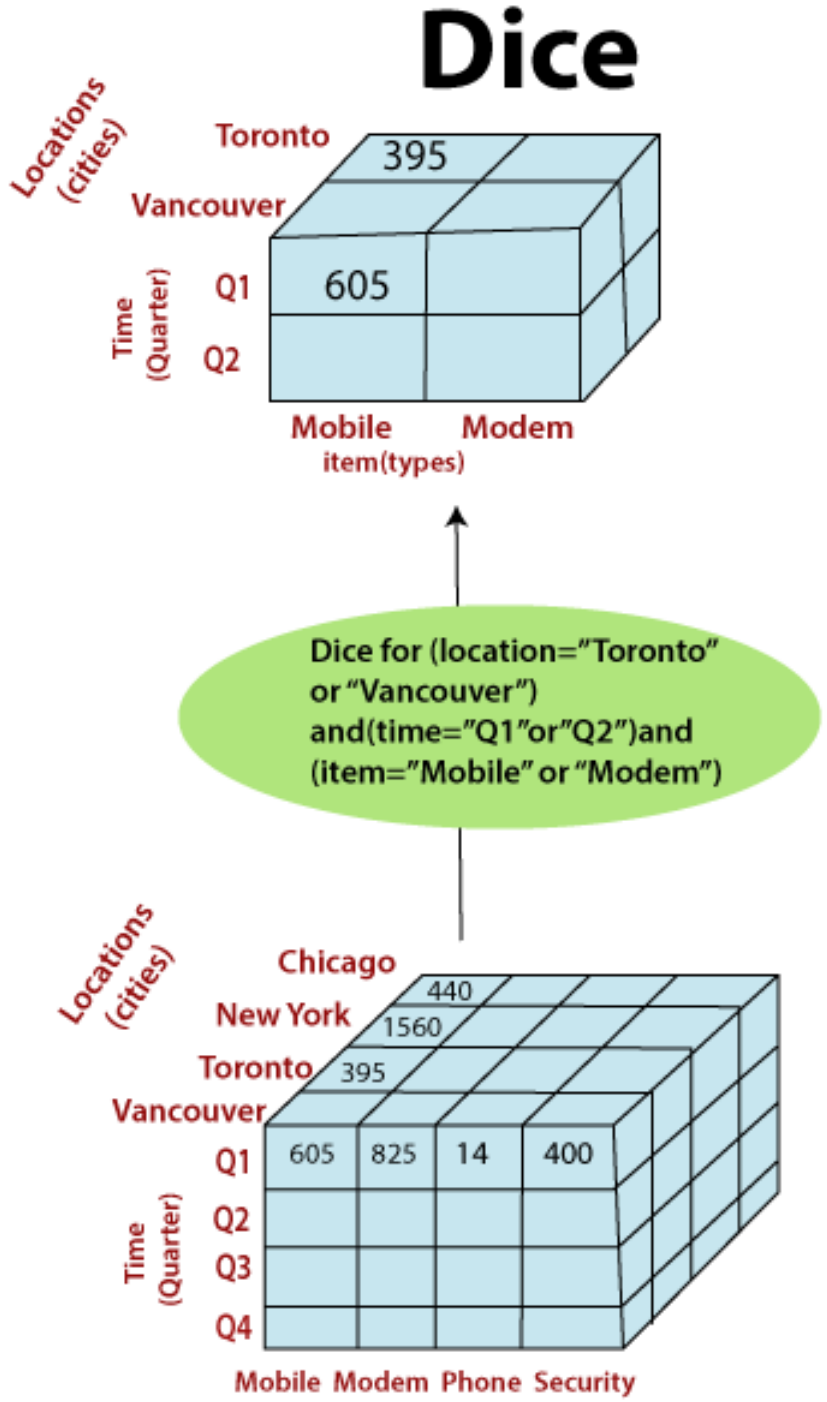
Operações OLAP - Slicing

Selecionar um valor **único** para uma das dimensões, criando um cubo com uma dimensão a menos.



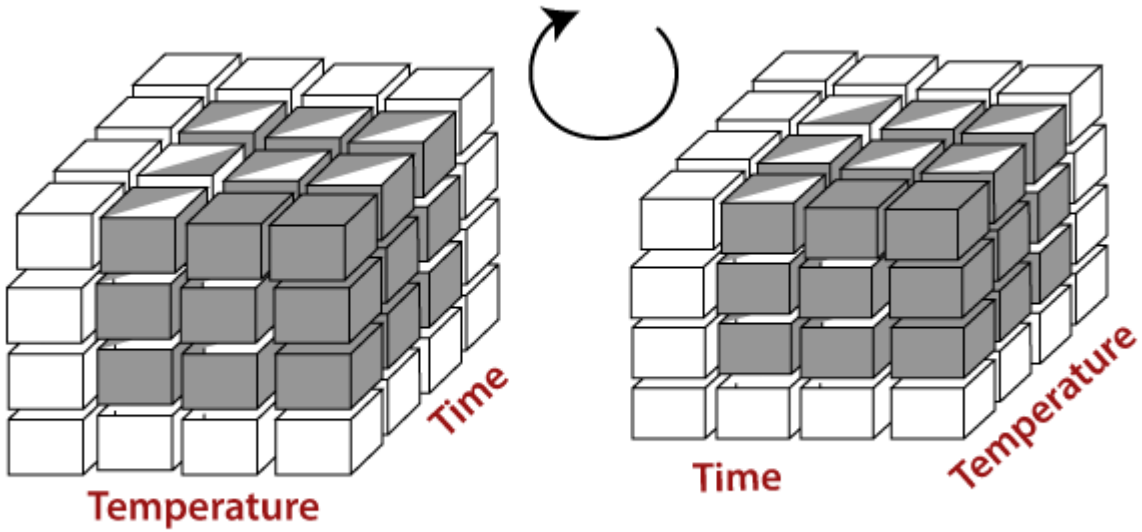
Operações OLAP - Dicing

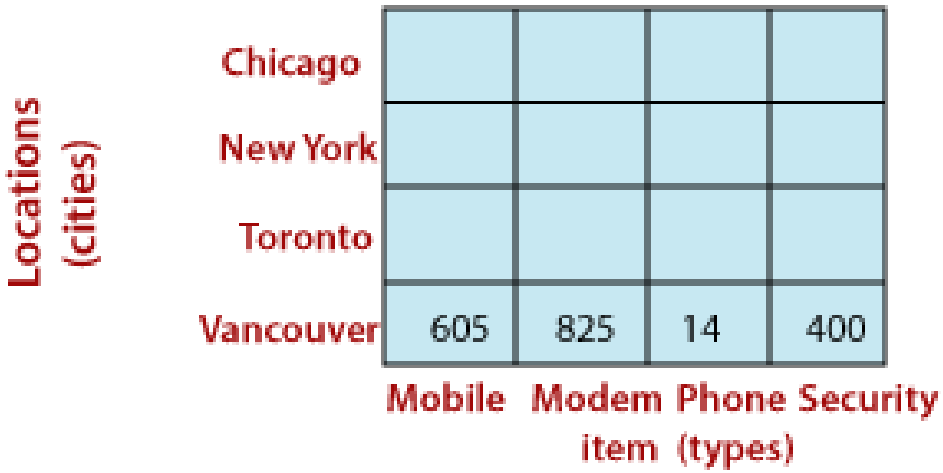
Produz um subcubo selecionando **valores** ou **intervalos** específicos em **uma** ou **múltiplas** dimensões.



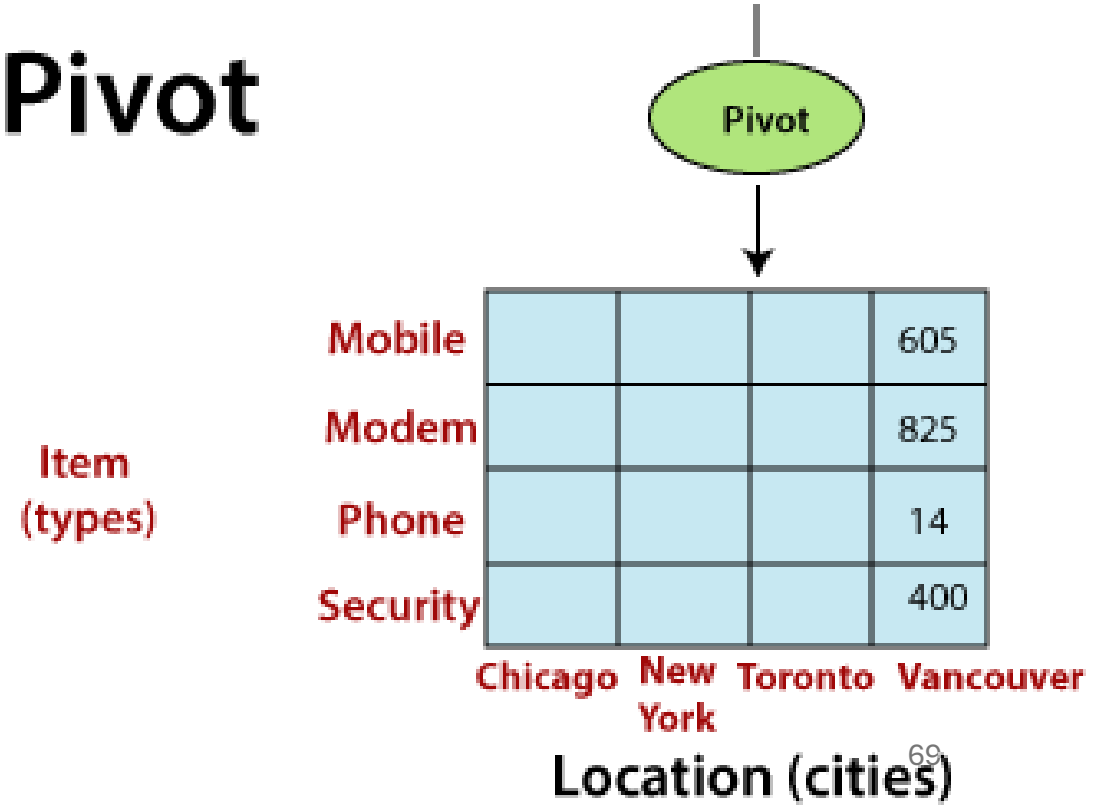
Operações OLAP - Pivoting

Rotacionar um cubo em um dos eixos – modificando a ordem das dimensões na análise visual





Pivot



Total de cuboids

- Quantos cuboids formam um cubo n-dimensional?
- **Caso 1:** Não há hierarquias relacionadas com as dimensões

$$2^n \text{cubóides}$$

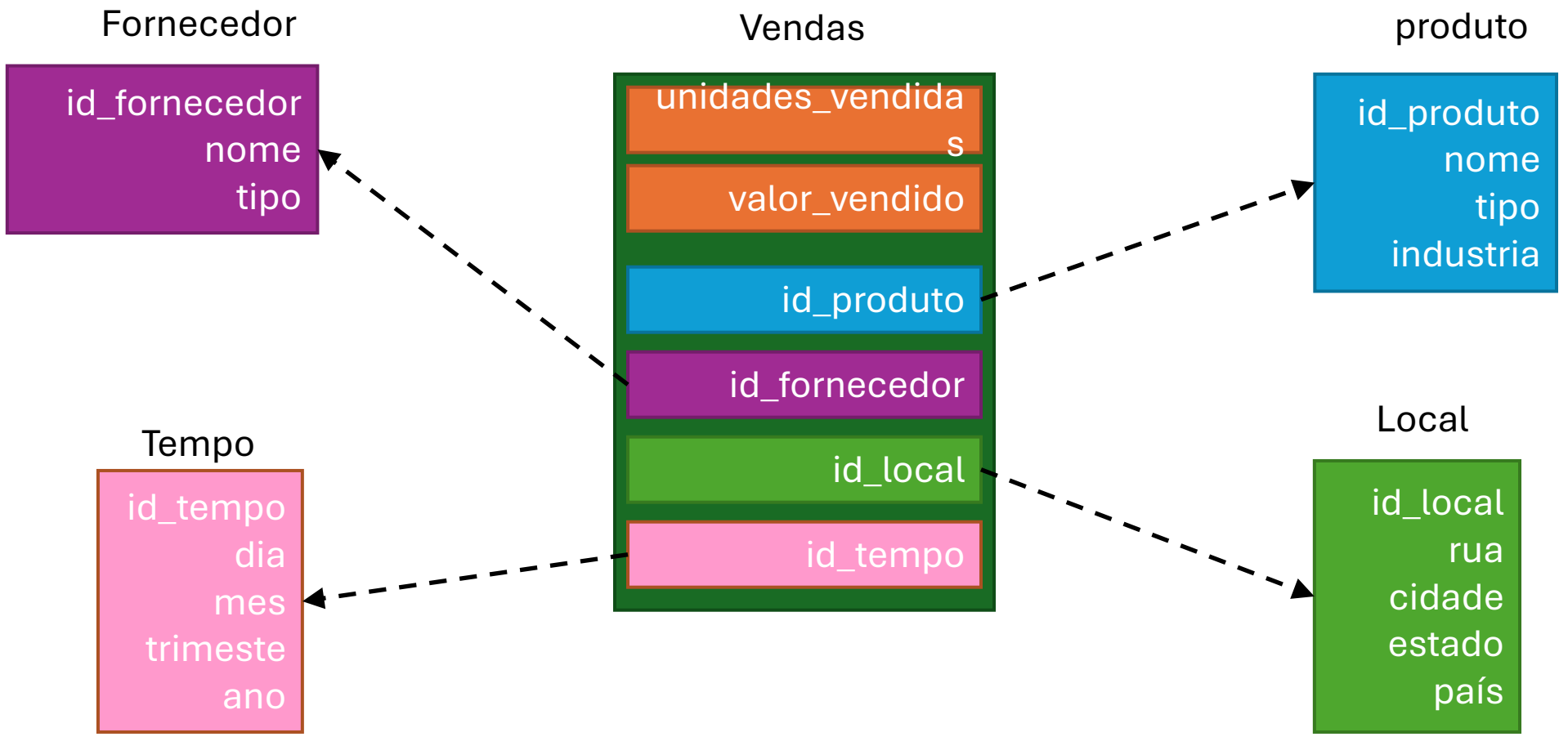
- **Caso 2**: Há hierarquias de conceitos

$$T = \prod_{i=1}^{n} (L_i + 1)$$

Onde L_i corresponde ao número de níveis na dimensão i

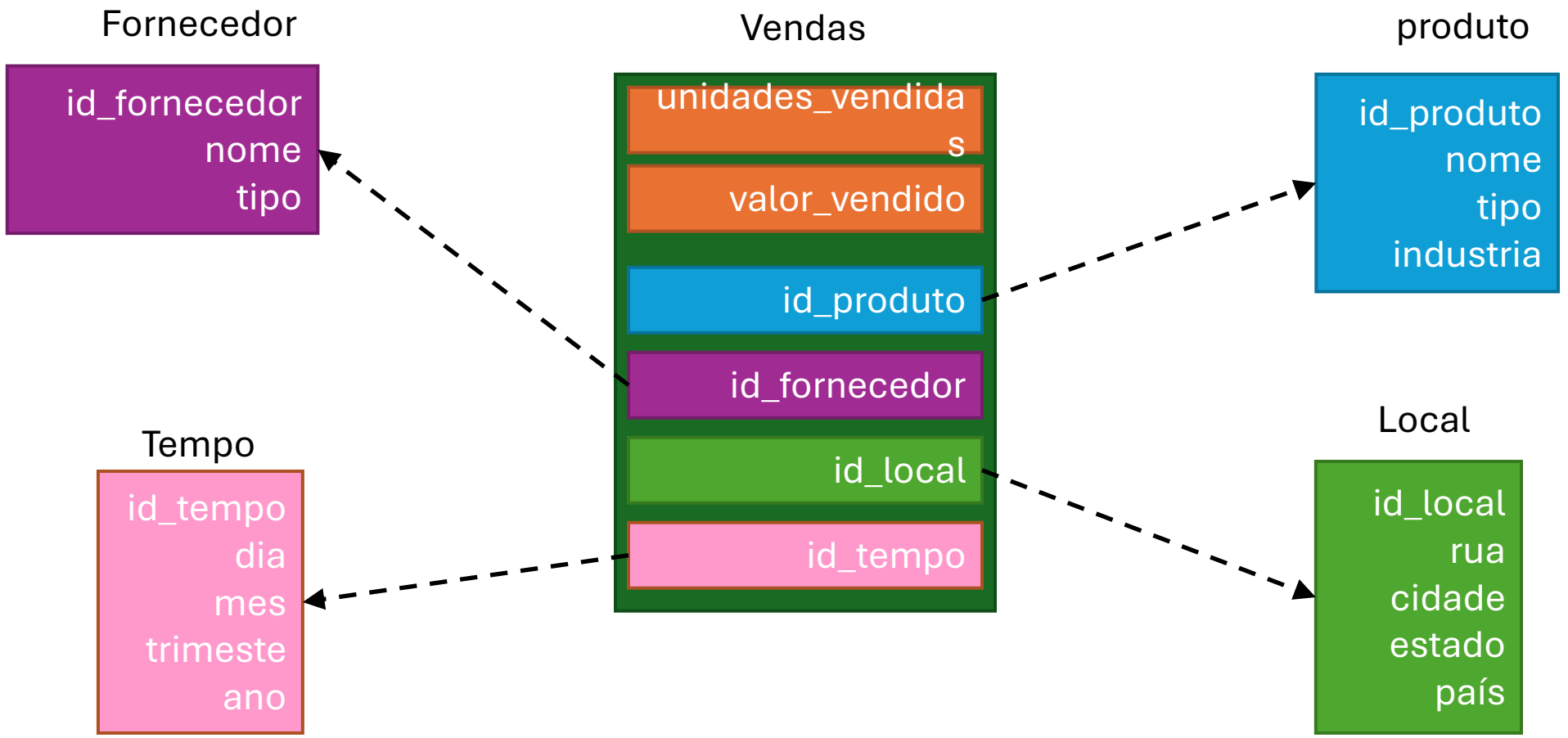
Total de cuboids

$$T = \prod_{i=1}^{n} (L_i + 1)$$



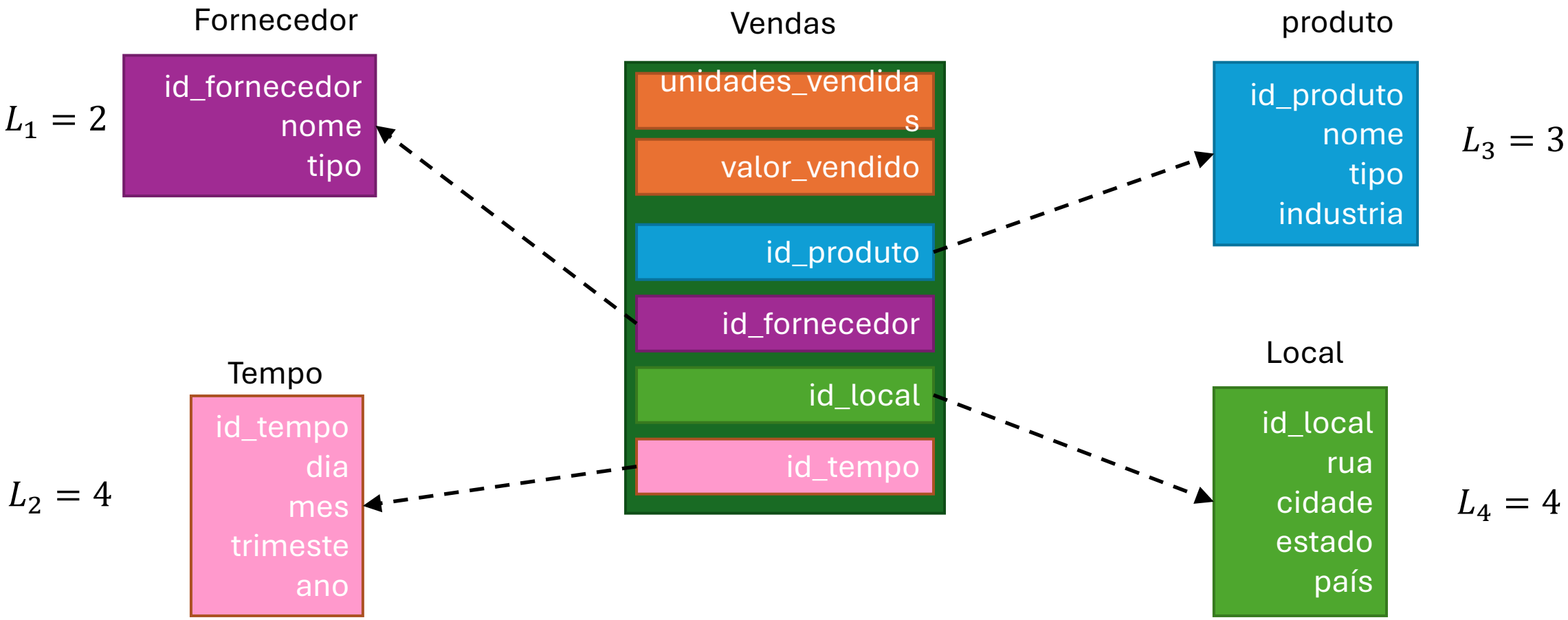
Total de cuboids

$$T = \prod_{i=1}^{n} (L_i + 1) \qquad n = 4$$



Total de cuboids

$$T = \prod_{i=1}^{n} (L_i + 1) \qquad n = 4$$

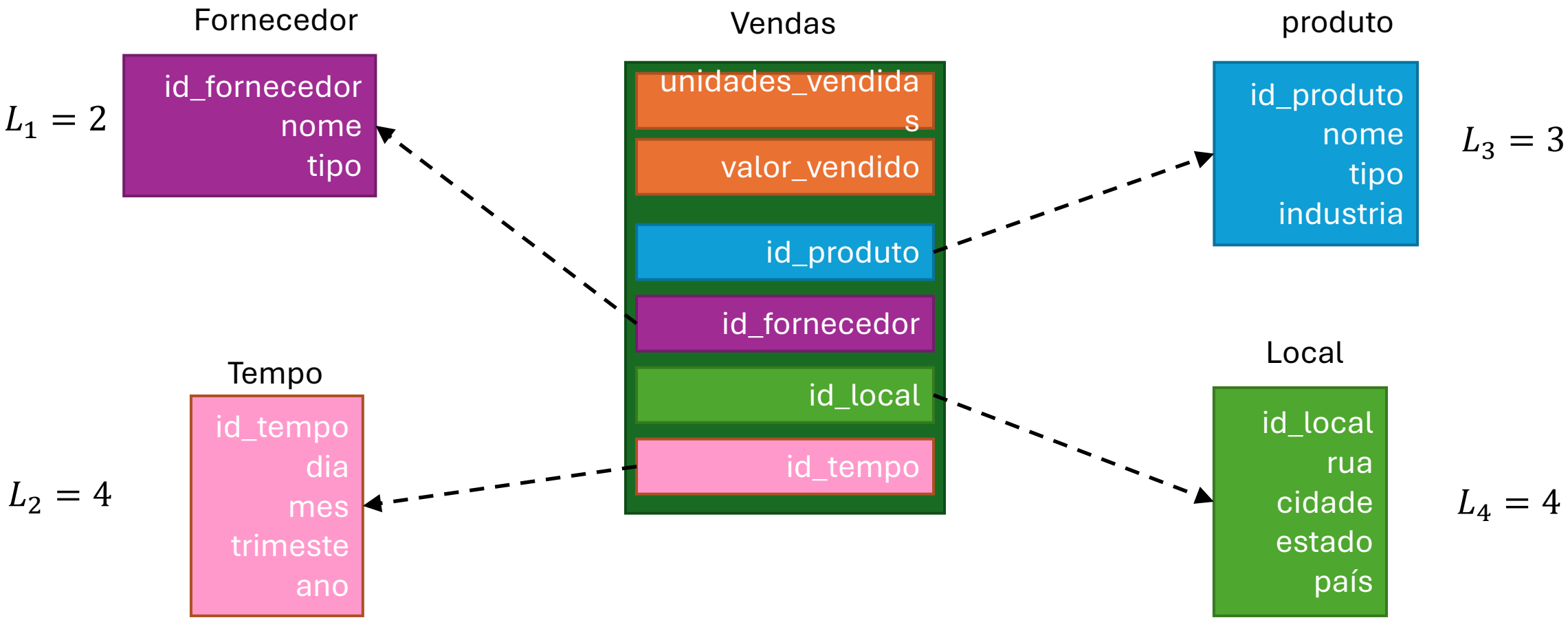


Total de cuboids

$$T = \prod_{i=1}^{n}(L_i + 1)$$

$n = 4$

Total de 300 cuboids



Exemplos de Cubos

Visão da Tabela Fato:

sale	prodId	storeId	amt
	p1	c1	12
	p2	c1	11
	p1	c3	50
	p2	c2	8

Cubo:

	c1	c2	c3
p1	12		50
p2	11	8	

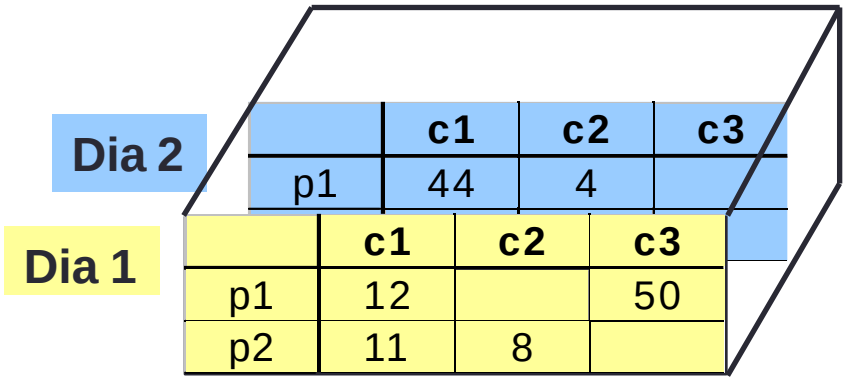
dimensões = 2

Exemplos de Cubos

Visão da Tabela Fato:

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

Cubo:



dimensões = 3

Materialização dos Cubos

- O cubo são todos os conjuntos de views possíveis da tabela de fatos e suas dimensões.
- Na prática, como já vimos, o armazenamento dos dados é feito nos schemas como Star, Snowflake, Constelação, etc.
- Quando precisamos de algum cuboid, precisamos **computar** ele de alguma forma:
  - Realizando agregações, seja pela soma dos valores (sum), contagem dos valores (Count), média, mediana, etc.
- Quando criamos de fato uma view destas, dizemos que estamos **materializando** o cubo, ou seja, computando algo que antes era apenas abstrato!

Materialização dos Cubos

- Nem sempre é viável pré-computar todas materializações possíveis de uma aplicação de larga escala, por isso separamos em 3 tipos de materializações.
  - **Sem materialização:** não pré- computamos nenhum dos cuboids que não sejam o cuboid base. Toda materialização precisa ser computada “on-the-fly”, o que pode ser bastante lento.
  - **Materialização completa:** pré-computar todos cuboids. Provavelmente será necessário muito espaço de armazenamento para guardar todas estas computações, que talvez nem sejam utilizadas.
  - **Materialização parcial:** pré-computar um subconjunto dos cuboids, de acordo com algum critério de um especialista do domínio, ou um subconjunto de células de um valor mínimo, etc. Geralmente é a escolha mais eficiente. A medida que views não computadas sejam requisitadas, computá-las e depois armazená-las para o futuro.

Referências

- Data Mining: Concept and Techniques
  - Capítulo 3