

# SISTEMAS DE INFORMAÇÃO II

## CAPÍTULO 1.3

### SISTEMAS DE DATA WAREHOUSING: INFRA-ESTRUTURA INFORMACIONAL E FUNCIONAL

FILIPPE SÁ  
FILIPPE.SA@ISEC.PT

## *1.3 Sistemas de Data Warehousing: Ciclo de Vida e Desenvolvimento*

### **Tópicos Programáticos:**

#### *1.5. Projecto, implementação e administração de um data warehouse*

##### *1.5.1. Modelação Multidimensional*

###### *1.5.1.1. Motivação*

###### *1.5.1.2. Introdução ao modelo multidimensional*

###### *1.5.1.3. Cubos, dimensões, factos e medidas*

###### *1.5.1.4. Modelação conceptual dum DW*

###### *1.5.1.5. Modelação – Esquema em Estrela (Star)*

###### *1.5.1.6. Modelação – Esquema em Floco-de-Neve*

###### *1.5.1.7. Modelação – Esquema em Constelação (Constellation)*

###### *1.5.1.8. Modelação – Qual é o melhor esquema?*

##### *1.5.2. Conceção de um DW: Processo em 4 Etapas*

###### *1.5.2.1. 1ª Etapa – Identificação dos Processos de Negócio*

###### *1.5.2.2. 2ª Etapa – Identificação da Granularidade*

###### *1.5.2.3. 3ª Etapa – Identificação das Dimensões*

###### *1.5.2.4. 4ª Etapa – Identificação dos Factos*

###### *1.5.2.5. Passos Adicionais*

###### *1.5.2.6. Calculo espaço para DW ou DM*

## *1.3 Sistemas de Data Warehousing: Ciclo de Vida e Desenvolvimento*

### **Tópicos Programáticos:**

#### *1.5. Projecto, implementação e administração de um data warehouse(cont.)*

##### *1.5.3. Considerações de concepção*

*1.5.3.1. Forças e Fraquezas do modelo multidimensional*

*1.5.3.2. Relacionamento muito-para-muitos*

*1.5.3.3. Dimensões e Factos Degenerados*

*1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”*

*1.5.3.5. Dificuldades na Implementação de um DW*

*1.5.3.6. Mitos Relativos à Modelação Dimensional*

*1.5.3.7. Ciladas a Evitar*

*1.5.3.8. Considerações Técnicas*

*1.5.3.9. Considerações de Implementação*

*1.5.3.10. Ferramentas e Utilitários Back-End*

*1.5.3.11. Soluções integradas*

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.1. Motivação: Porquê um novo modelo

- Estamos familiarizados com a modelação E/R e OO;
- Todos os tipos de dados são iguais;
- Os modelos E/R e OO são:
  - Flexíveis;
  - Gerais.
- Não há diferenciação possível para:
  - Especificar o que é realmente importante;
  - O que apenas descreve o que é importante.
- Os modelos ER/OO são grandes:
  - 50-1000 entidades / relações / classes;
  - Difícil de obter uma panorâmica geral.
- Os modelos ER/OO são implementados em RDBMSs
  - As bases de dados normalizadas “dispersam” informação;
  - Para análise, os dados têm de ser integrados de novo.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.2. Introdução ao Modelo Multidimensional

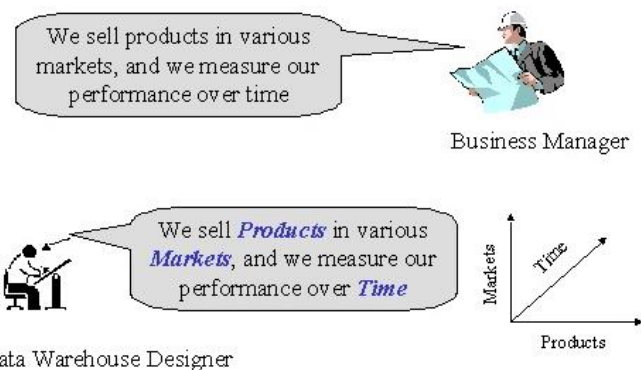
A modelagem multidimensional, ou dimensional como às vezes é chamada, é a técnica de modelação de base de dados para o auxílio às consultas do Data Warehouse nas mais diferentes perspetivas

O modelo multidimensional resulta, directamente, da necessidade sentida pelo decisor de poder **manipular** livremente os **dados** e **visualizá-los segundo a sua forma nativa de ver o negócio**.

O negócio é visto sob a forma **multi-perspectiva**, em que um dado facto pode ser analisado segundo **caracterizações várias**, que devem ser passíveis de manipulação directa pelo próprio utilizador.

Ou seja, o utilizador deve poder interagir directamente com o esquema lógico multidimensional do sistema multidimensional para formular consultas.

#### Describing the Organization



## *1.4. Projecto, implementação e administração de um DW*

### *1.5.1. Modelação Multidimensional*

#### *1.5.1.2. Introdução ao Modelo Multidimensional*

A modelação dimensional pode parecer estranha aos profissionais das TIC, familiarizados com a modelação relacional, pois começa com tabelas em vez de entidades relacionamento.

Estas tabelas podem ser de dois tipos, Factos ou Dimensões.

As tabelas de Factos contêm valores, enquanto as tabelas de Dimensão contêm medidas dos valores contidos nas tabelas de Factos.

As tabelas de Factos contêm grupos repetidos, o que viola as regras da normalização do modelo relacional.

Contudo, esta violação das regras da normalização tem como objetivo aumentar o desempenho do DW

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.1. Modelação Multidimensional

#### *1.5.1.2. Introdução ao Modelo Multidimensional*

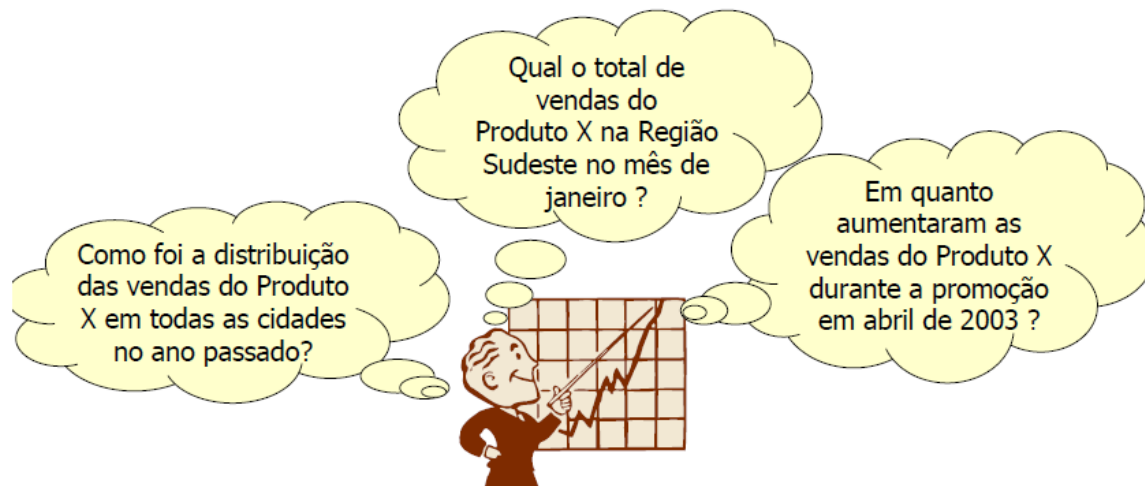
- Um propósito:
  - análise de dados
- Mais adequado para o propósito analítico
  - menos flexível;
  - não adequado para sistemas OLTP.
- Tem embutido o próprio “significado”
  - o que é importante;
  - o que descreve o que é importante;
  - o que queremos otimizar;
  - agregações automáticas significam fácil interrogação.
- Reconhecido pelas ferramentas OLAP/BI
  - as ferramentas oferecem facilidades de interrogação poderosas baseadas no desenho multidimensional.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.2. Introdução ao Modelo Multidimensional

Forma pela qual os analistas de negocio, gestores e executivos analisam informações



Objectivos da modelação multidimensional:

- rodear os factos com tanto contexto (dimensões) quanto possível;
- dica: a redundância pode ser boa (em locais bem escolhidos);
- mas não se deve tentar modelar todos os relacionamentos nos dados (tal como na modelação E/R e OO).



## *1.4. Projecto, implementação e administração de um DW*

### 1.5.1. Modelação Multidimensional

#### *1.5.1.2. Introdução ao Modelo Multidimensional*

Foco no cruzamento de informações

Facilita o entendimento e visualização de problemas típicos de suporte à decisão

Mais intuitiva para o processamento analítico

Utilizada pelas ferramentas OLAP

É baseada em dois pressupostos:

- produzir uma estrutura da base de dados fácil de compreender e utilizar (facilitando a colocação de interrogações ao sistema)
- otimizar o desempenho no processamento de questões, em oposição à otimização do processamento de atualizações, como se verifica no modelo relacional das base dados

## 1.4. Projecto, implementação e administração de um DW

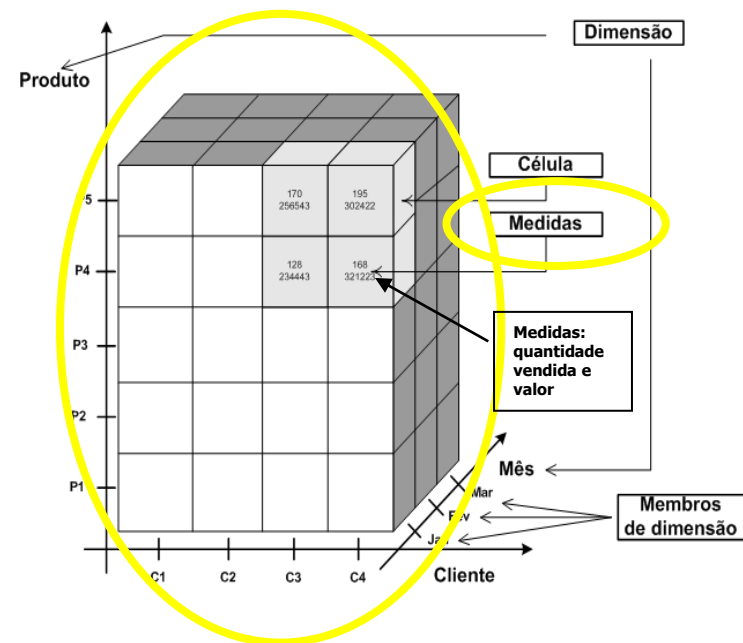
### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. *Cubos, Dimensões, Factos e Medidas*

A terminologia de modelo multidimensional provém do facto de este modelo poder ser visualizado como um cubo, no qual cada uma das suas faces caracteriza uma tabela de dimensões do modelo em estrela

Um **cubo de dados**, em termos técnicos, não é mais do que uma projecção multidimensional redundante de uma relação, tendo sido proposto em [Gray et al., 1996] como uma generalização do operador SQL Group-By;

Um cubo pode ser visualizado como uma **grelha multidimensional** construída a partir dos valores das dimensões;



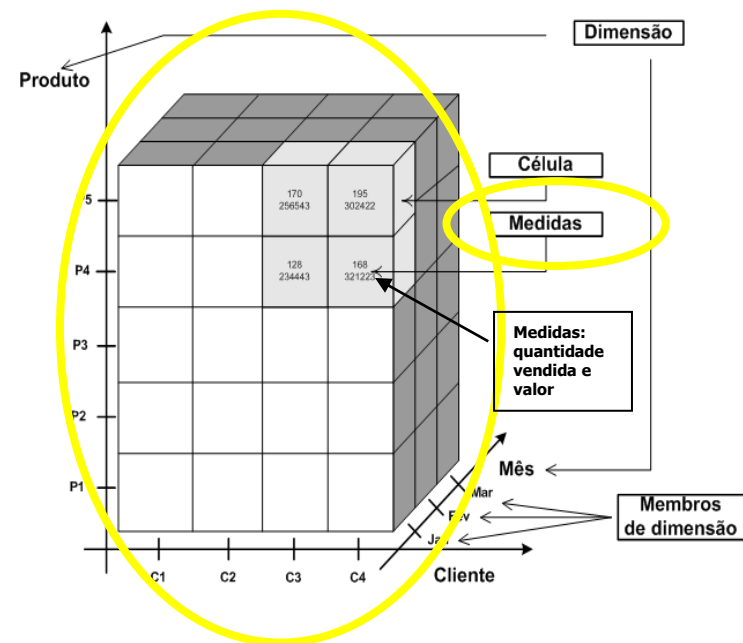
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. *Cubos, Dimensões, Factos e Medidas*

Cada **célula** nesta grelha contém uma série de **medidas** (valores numéricos que “vivem” dentro da célula), caracterizados pela **mesma combinação de coordenadas**, instância para dimensões ou níveis e medem um **facto** do negócio;

Os **factos** mais úteis são **numéricos e aditivos**, e.g., total de vendas.

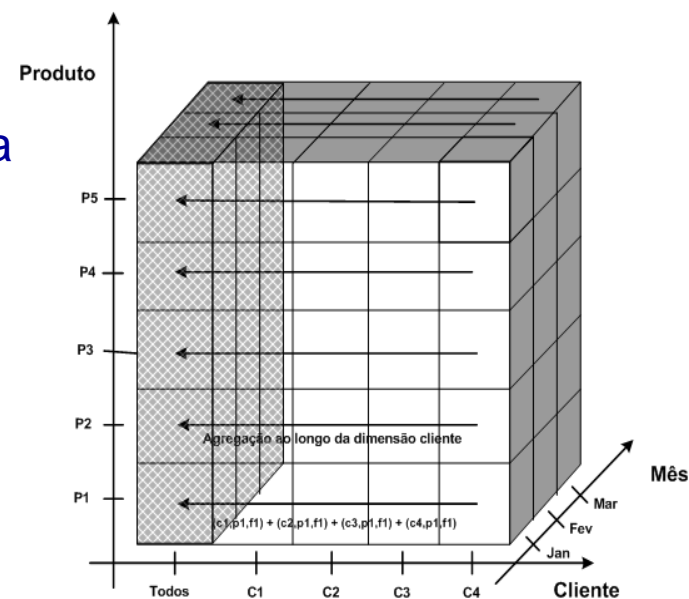


## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, Factos e Medidas

- Um **cubo** pode ter muitas dimensões
  - se mais de 3, o termo "hipercubo" é usado muitas vezes;
  - teoricamente, não há limite para o número de dimensões;
  - cubos típicos têm entre 4 a 12 dimensões.
- Mas só 2-3 dimensões podem ser visualizadas de cada vez
  - a dimensionalidade é reduzida através de operações de projecção/agregação (figura ao lado).
- Um cubo consiste em células
  - uma dada combinação de valores de dimensões;
  - uma célula pode estar vazia (não haver dados para a combinação de valores das dimensões);
  - um cubo disperso tem poucas células não vazias (com valores);
  - um cubo denso tem muitas células não-vazias;
  - os cubos tendem a tornar-se dispersos para muitas/grandes dimensões.



## 1.5. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. *Cubos, Dimensões, Factos e Medidas*

Visão Relacional

Volume de vendas para o  
concessionário xpto

Visão multidimensional

MODEL	CITY	SALES VOLUME
MINI VAN	NEW YORK	6
MINI VAN	LOS ANGELES	5
MINI VAN	MADISON	4
SPORTS COUPE	NEW YORK	3
SPORTS COUPE	LOS ANGELES	5
SPORTS COUPE	MADISON	5
SEDAN	NEW YORK	4
SEDAN	LOS ANGELES	3
SEDAN	MADISON	2

MODEL	Mini Van	6	5	4
	Coupe	3	5	5
	Sedan	4	3	2
		NY	LA	Madison
		CITY		

Um vetor multidimensional tem um número fixo de dimensões e os valores são armazenados nas células  
Cada dimensão consiste de um número de elementos

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

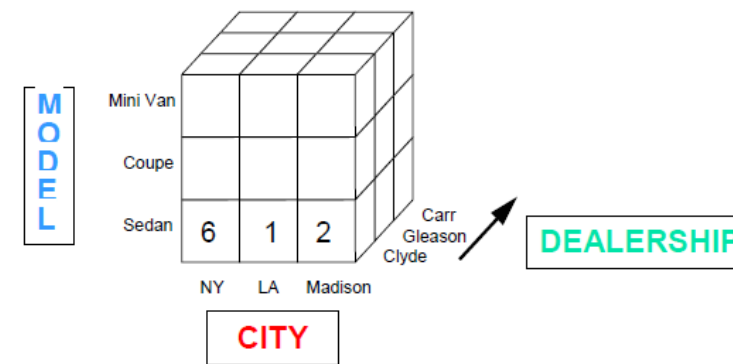
#### 1.5.1.3. Cubos, Dimensões, Factos e Medidas

Acrescentando mais uma coluna

Volume de vendas para o  
concessionário xpto

Visão multidimensional

MODEL	CITY	DEALERSHIP	VOLUME
MINI VAN	NEW YORK	CLYDE	6
MINI VAN	NEW YORK	GLEASON	6
MINI VAN	NEW YORK	CARR	2
MINI VAN	LOS ANGELES	CLYDE	3
MINI VAN	LOS ANGELES	GLEASON	5
MINI VAN	LOS ANGELES	CARR	5
MINI VAN	MADISON	CLYDE	2
MINI VAN	MADISON	GLEASON	4
MINI VAN	MADISON	CARR	3
SPORTS COUPE	NEW YORK	CLYDE	2
SPORTS COUPE	NEW YORK	GLEASON	3
SPORTS COUPE	NEW YORK	CARR	2
SPORTS COUPE	LOS ANGELES	CLYDE	7
SPORTS COUPE	LOS ANGELES	GLEASON	5
SPORTS COUPE	LOS ANGELES	CARR	2
SPORTS COUPE	MADISON	CLYDE	4
SPORTS COUPE	MADISON	GLEASON	5
SPORTS COUPE	MADISON	CARR	1
SEDAN	NEW YORK	CLYDE	6
SEDAN	NEW YORK	GLEASON	4
SEDAN	NEW YORK	CARR	2
SEDAN	LOS ANGELES	CLYDE	1
SEDAN	LOS ANGELES	GLEASON	3
SEDAN	LOS ANGELES	CARR	4
SEDAN	MADISON	CLYDE	2
SEDAN	MADISON	GLEASON	2
SEDAN	MADISON	CARR	3



Dados podem ser imaginados como um “cubo”

- metáfora visual
- representação intuitiva: dimensões coexistem para todo ponto no cubo e são independentes umas das outras

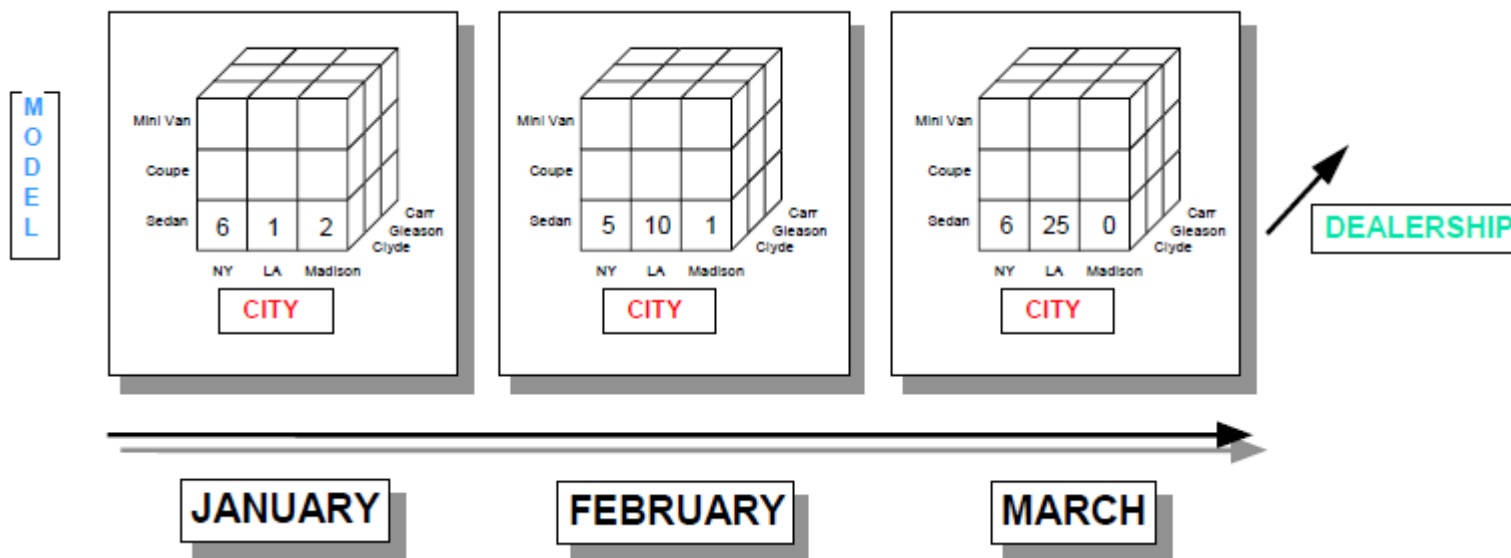
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. *Cubos, Dimensões, Factos e Medidas*

Volume de vendas para o concessionário xpto,  
ao longo do tempo

Nota: Os cubos serão apresentados no capítulo  
sobre OLAP



## 1.5. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, *Dimensões*, Factos e Medidas

Os dados são divididos em Factos e Dimensões

Existem tantas dimensões quantas as vertentes pelas quais se pretende analisar os factos

Apresentam-se não normalizadas, integrando na maioria dos casos uma grande quantidade de atributos;

Contêm poucos registos quando comparadas com a tabela de factos, apesar de integrarem muitos atributos;

- **Dimensões:**

- Chave simples
- Fonte principal das cláusulas das consultas, agrupamentos e títulos de relatórios
- Volume de vendas “por produto”
- Usualmente não dependente do tempo
- Desnormalizada (em principio)
- Hierarquias implícitas

#### Product Dimension

product\_key  
description  
brand  
category  
department  
package type  
package size  
fat content  
diet type  
weight  
weight unit of measure  
storage type  
...



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, *Dimensões*, Factos e Medidas

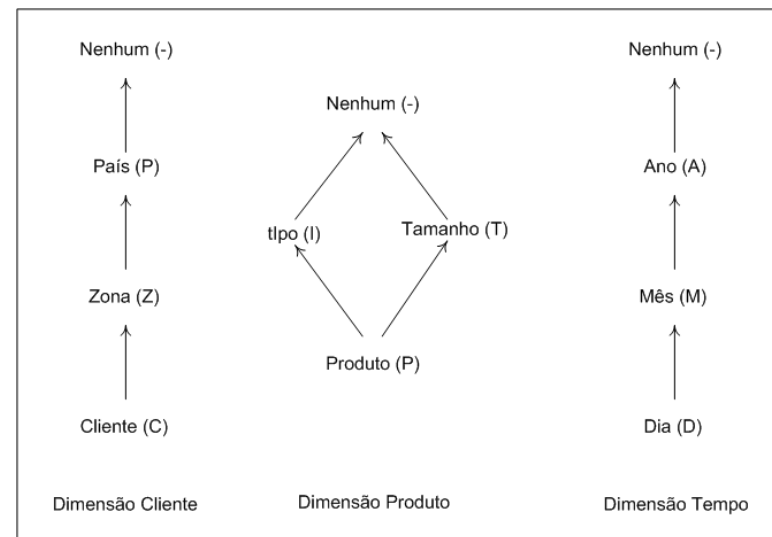
A noção de **dimensão** é um conceito essencial e diferenciador em dados multidimensionais.

As dimensões são utilizadas com dois propósitos: a selecção de dados e o seu agrupamento a um dado nível.

Cada dimensão está organizada sob a forma de uma ou várias hierarquias.

Na figura ao lado: a hierarquia da dimensão produto poderia ser do tipo:

- produto → tipo ou
- produto → tamanho, sendo o produto, neste caso, uma dimensão múltipla.



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, *Dimensões*, Factos e Medidas

Cada hierarquia é composta de um número de níveis, cada um representando um nível de detalhe que pode interessar às análises a ser executadas.

Para cada par dimensão/nível, teremos um conjunto de instâncias:

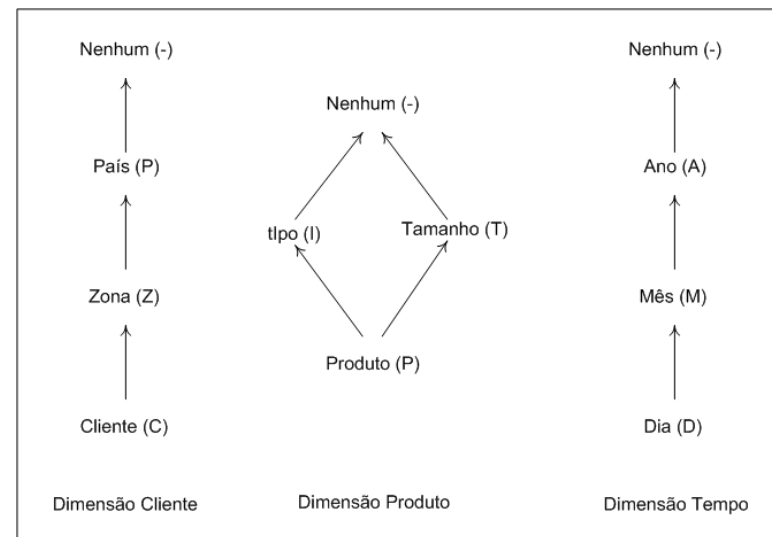
- para mês, ter-se-á Janeiro, Fevereiro, etc.
- para zona geográfica, a zona norte, centro, etc.

Denominam-se tipicamente por membros de dimensão ou valores de dimensão.

As dimensões devem conter muita informação:

- A dimensão tempo pode conter férias, estação, eventos, ...
- Boas dimensões têm 50-100 ou mais atributos/níveis.

O nível das dimensões ao qual ocorre a combinação de valores irá determinar a **granularidade** do facto. No exemplo do cubo do slide 12, a granularidade do cubo é “produto por cliente por mês” (cpm).



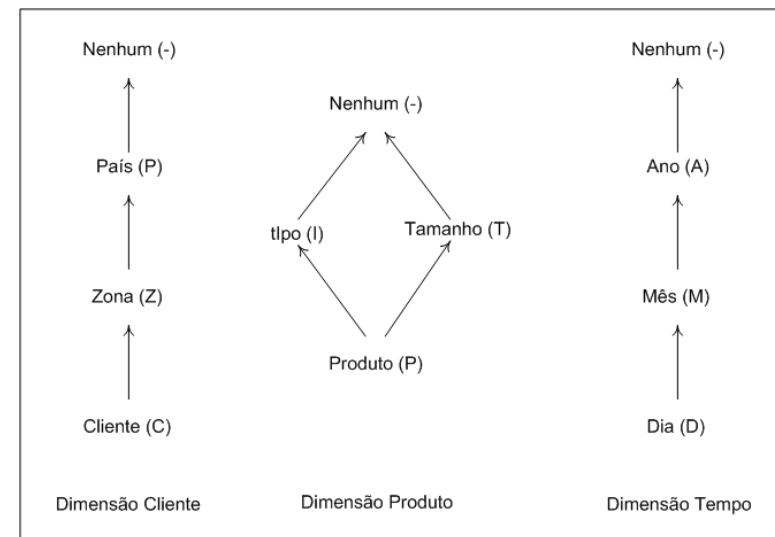
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, *Dimensões*, Factos e Medidas

Definir a granularidade é o processo de escolha do detalhe dos dados contidos no DW. O nível mais baixo de granularidade é designado de atómico, significando que não pode ser dividido.

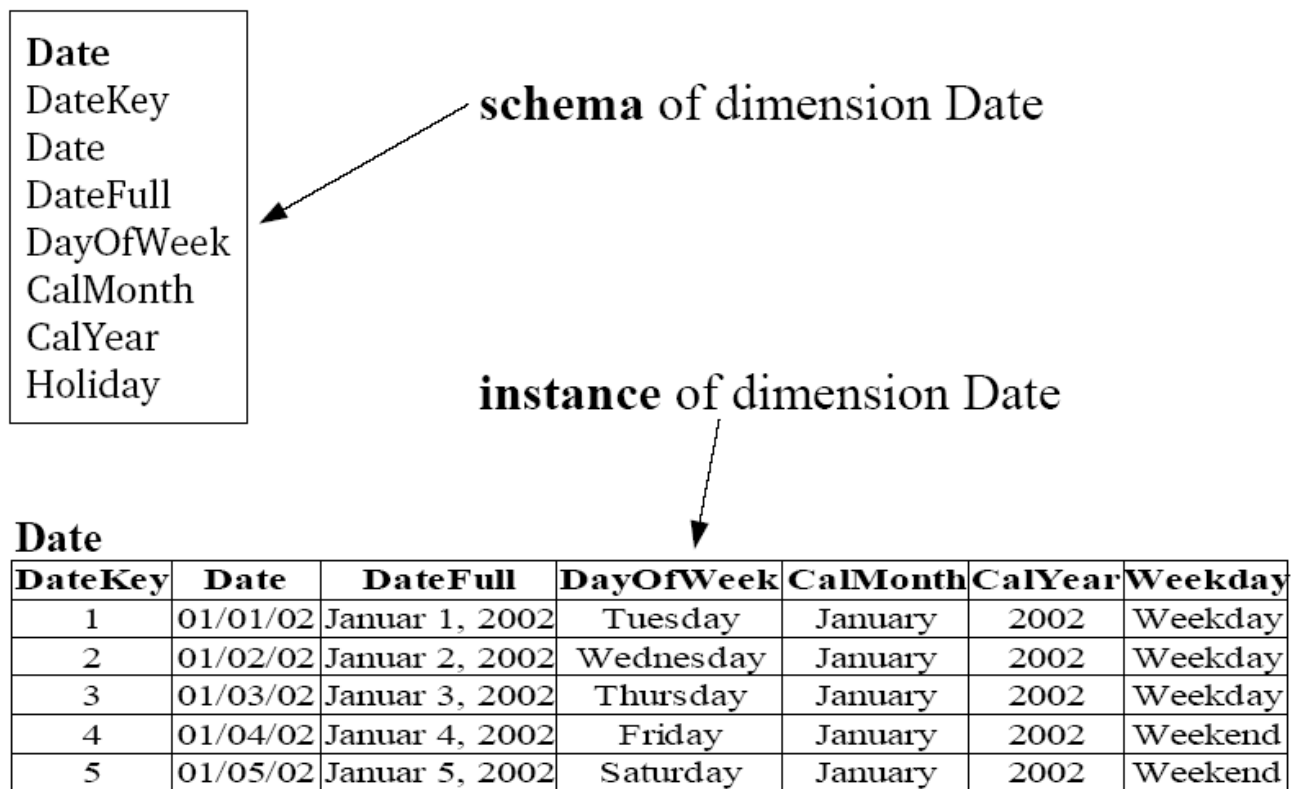
Escolher um nível de grão atómico é desejável, porque permite aos utilizadores agregar os dados à sua vontade, enquanto que escolher um nível intermédio de grão implica o risco de não ser possível satisfazer todas as pesquisas solicitadas pelos utilizadores.



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, *Dimensões*, Factos e Medidas



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, *Dimensões*, Factos e Medidas

- **Tabela de Factos:**
- **Integra um conjunto de atributos numéricos (factos) e um conjunto de chaves estrangeiras que relacionam a tabela factos com a tabela dimensões**
- **Apresentam-se normalizadas**
- **Contem um grande quantidade de registos, ocupando normalmente mais de 95% do espaço ocupado pela Data Warehouse.**
- **Contém as métricas. Possuem o carácter quantitativo das informações descritivas armazenadas nas Dimensões. É onde estão armazenadas as ocorrências do negócio e possui relacionamento de “muitos para um” com as tabelas periféricas (Dimensão).**

## 1.5. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, *Factos* e Medidas

Tabela de Factos:

- são a entidade importante, e.g., volume vendas, volume transações de cartão de crédito, etc;
- têm medidas que podem ser agregadas;
- vivem num cubo multidimensional que pode ser visto como um array.
- Quantidade de registos é normalmente muito grande

SGBD relacional

- Geralmente são a tabela Relação (Tabela)
- Chaves estrangeiras para as tabelas de dimensão
- Chave primária é subconjunto das chaves estrangeiras

SGBD multidimensional

- Cubo (vetor n-dimensional)

**As tabelas de Factos contêm muitas linhas e poucas colunas. Isto é essencial para facilitar o uso e aumentar o desempenho**

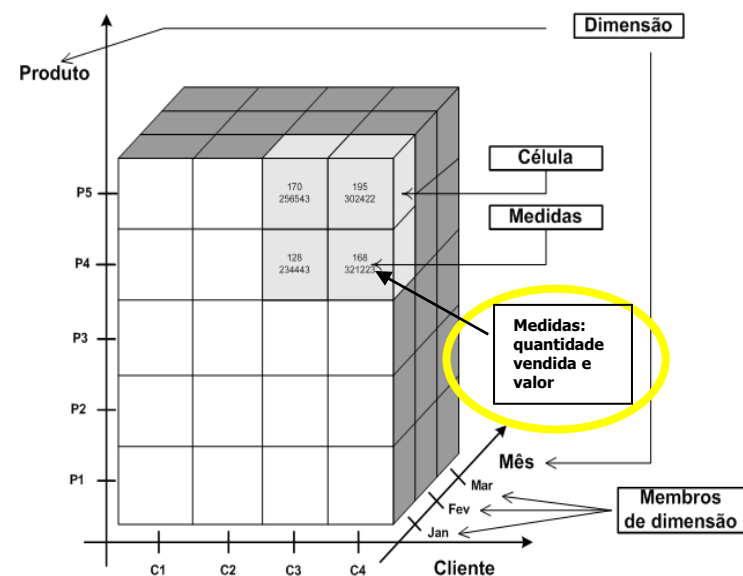
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, *Factos* e Medidas

### Factos

- Factos representam o assunto da análise desejada
  - a coisa importante do negócio que deve ser analisada
- Um facto é identificado pelas valores nas suas dimensões
  - um facto é uma célula não vazia
  - alguns modelos dão aos factos uma identidade explícita
- Geralmente um facto deve:
  - estar ligado a exactamente um valor em cada dimensão
  - estar ligado somente a valores de dimensão dos níveis mais baixos
  - alguns modelos não requerem isto



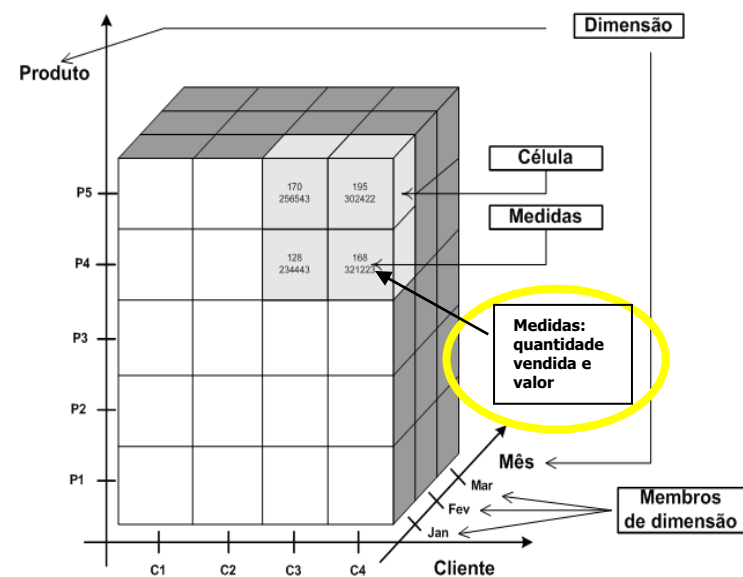
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, *Factos* e Medidas

##### Factos

- **Facto Evento (transacção)**
  - Um facto para cada evento do negócio (venda)
- **Facto "Fact-less"**
  - um facto por evento (contacto com cliente)
  - medidas não numéricas
  - um evento aconteceu para uma combinação de valores para as dimensões
- **Facto Snapshot**
  - um facto para cada combinação de valores em dimensões num dado intervalo de tempo
  - captura o estado corrente (ex. inventário)





## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, *Factos* e Medidas

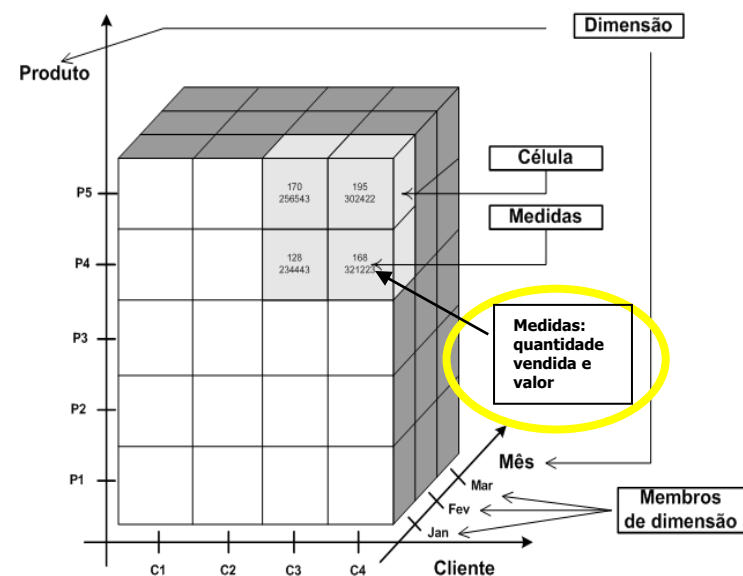
#### Factos (cont.)

- **Facto Snapshot Cumulativo**

- um facto para cada combinação de valores nas dimensões num dado intervalo de tempo
- captura o estado cumulativo até ao momento (vendas até à data)

- **Cada tipo de facto responde a diferentes questões**

- muitas vezes existem factos eventos e factos snapshot



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, Factos e Medidas

### Medidas

As medidas representam a propriedade do facto que o analista quer estudar e otimizar

- Exemplo: valor total de vendas

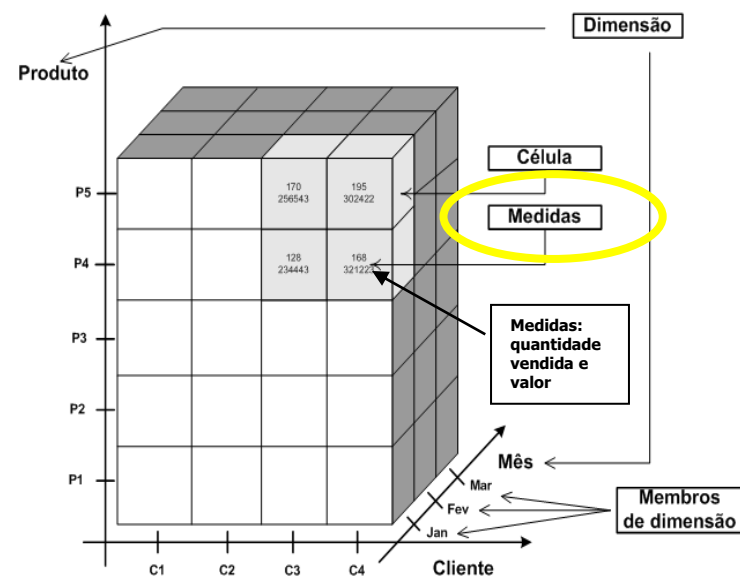
Uma medida tem dois componentes:

- O valor numérico (valor vendas);
- A fórmula de agregação (SUM): usada para agregação / combinação de um número de medidas.

O valor da medida é determinado pela combinação de valores da dimensão

O valor da medida tem significado para qualquer nível de agregação

A maioria dos modelos multidimensionais têm medidas



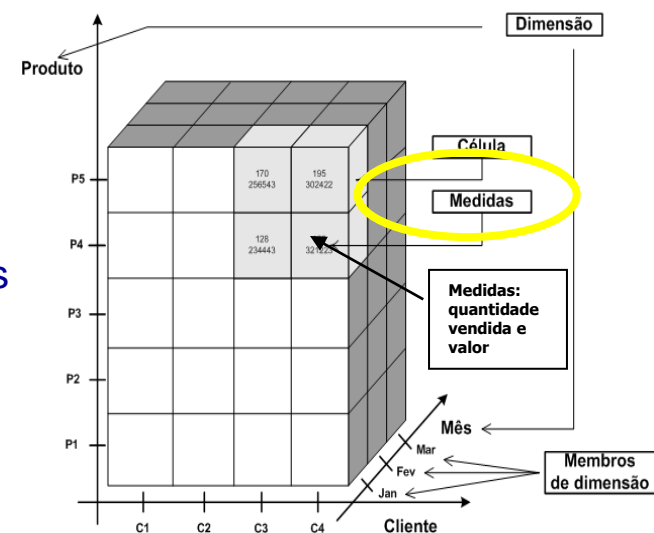
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.3. Cubos, Dimensões, Factos e *Medidas*

##### Medidas (tipos):

- **Aditivas**
  - Podem ser agregadas em todas as dimensões
  - Exemplo: valor da venda
  - Ocorrem, na generalidade, em factos do tipo eventos
- **Semi-aditivas**
  - Não podem ser agregadas em algumas dimensões – tipicamente as ligadas com o tempo- podem ser somadas, mas há que tomar cuidado para não cometer erros
  - Exemplo: quantidade em stock
  - Muitas vezes são factos derivados ou ocorrem em factos *snapshot*
- **Não aditivas**
  - Não podem ser agregadas em nenhuma dimensão
  - Exemplo: margem bruta ou número de funcionários
  - Ocorrem em todos os tipos de factos



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

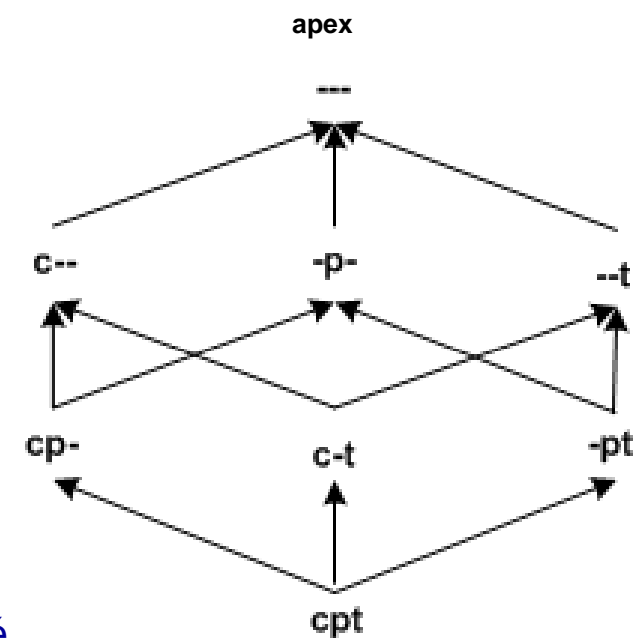
#### 1.5.1.3. Cubos, Dimensões, Factos e Medidas (resumo)

Um data warehouse é baseado num modelo de dados multidimensional que vê os dados sob a forma de um **cubo de dados**

Um cubo de dados, tal como vendas, permite que os dados sejam modelados e visualizados segundo múltiplas dimensões

- Tabelas **dimensão**, tal como cliente (código, zona, país) ou tempo (dia, semana, mês, trimestre, ano);
- A tabela **factos** contém medidas (como valor de vendas) e chaves para cada uma das tabelas dimensões relacionadas.

Na literatura relativa a data warehousing, um n-D cubo base é denominado **cubóide ou subcubo base**. O subcubo do topo 0-D, representa o nível mais elevado de agregação e é denominado o **subcubo apex**. O lattice de subcubos forma um **cubo de dados**.



Cubo de Dados (3D)

## 1.5. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.4. Modelação conceptual de um Data Warehouse

Segundo Kimball há que considerar o seguinte princípio:

*“Os modelos entidade-relacionamento típicos de OLTP são um desastre para consultas, porque não podem ser compreendidos pelo utilizador, e não podem ser navegados de forma útil por software DBMS. Esses modelos E-R não podem ser utilizados como base para os DW das empresas”.*

- Para o desenvolvimento do modelo físico do DW, ter-se-á de levar em conta os requisitos analíticos da aplicação, materializados no modelo dimensional, dependente do processamento analítico a efectuar.
- Mais ainda: o modelo físico resulta das modificações necessárias para atingir os níveis de desempenho convenientes, já que o seu esquema determina o desempenho, mas também limita o alcance do projecto.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.4. Modelação conceptual de um Data Warehouse

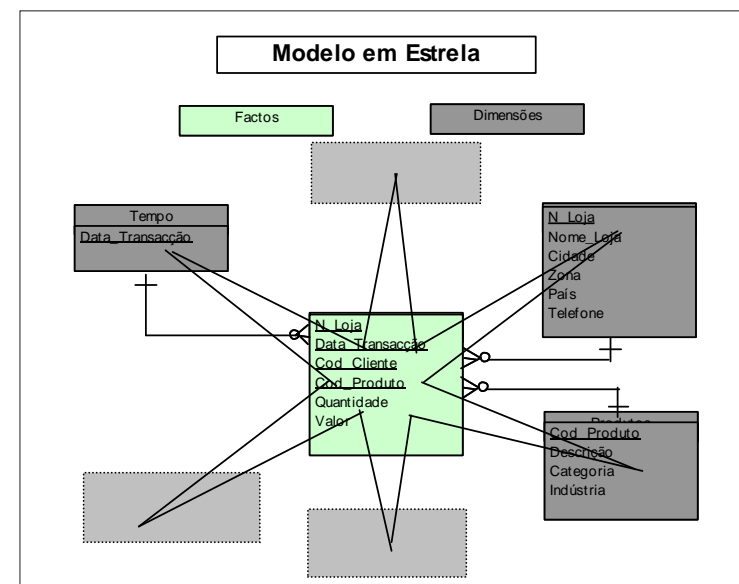
Modelar um Data Warehouse:

Dimensões e Medidas

Esquema em Estrela: Uma tabela factos no centro, ligada a um conjunto de tabelas dimensão;

Esquema em Floco-de-Neve: Um refinamento do esquema em estrela, onde cada hierarquia dimensional é normalizada num conjunto de tabelas dimensão menores, tendo um aspecto de um floco de neve;

Constelação de Factos: Múltiplas tabelas facto que partilham tabelas dimensão, vista como uma colecção de estrelas, então chamada esquema em galáxia ou constelação de factos.



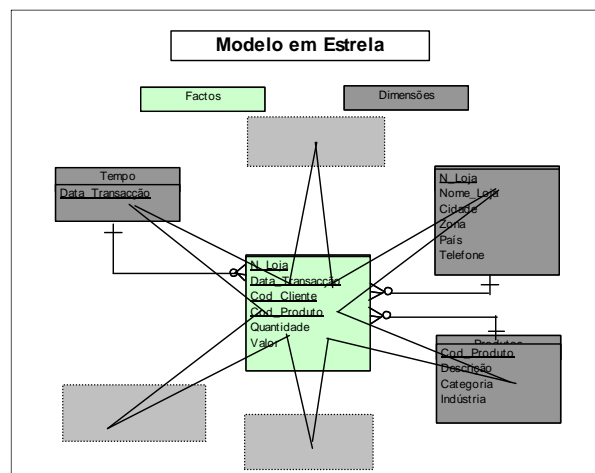
## 1.5. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela

A forma mais comum de modelar dados para uso dimensional é através do esquema em estrela (Star Schema)

Enquanto um esquema relacional de BD é projetado para ser eficiente no armazenamento de dados provenientes das transações, evitando a redundância, o esquema em estrela integra uma única tabela de factos, que constitui o centro da estrela, e múltiplas tabelas de dimensão ligadas à tabela de factos



Fonte Santos & Ramos (2017)

## 1.4. Projecto, implementação e administração de um DW

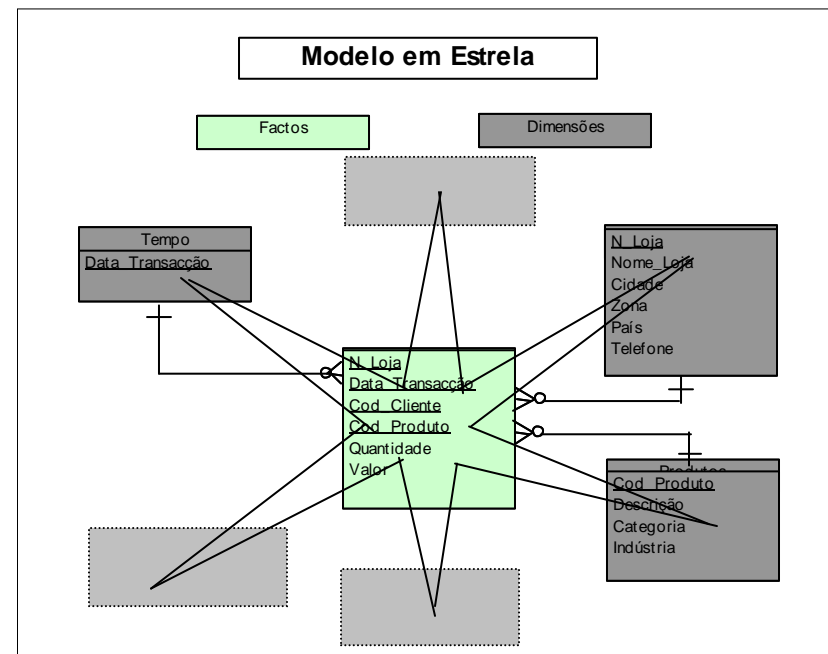
### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela

**Esquema em Estrela:** Uma tabela factos no centro, ligada a um conjunto de tabelas dimensão

- À volta de uma **tabela central**, onde serão **registados os valores** sobre os **factos** em análise (daí o chamar-se tabela factos),
- Um **conjunto de outras tabelas**, que proporcionam as **perspectivas** (dimensões) de avaliação dos factos.

Temos assim **uma tabela central e várias outras à sua volta**, constituindo as **pontas da estrela**.





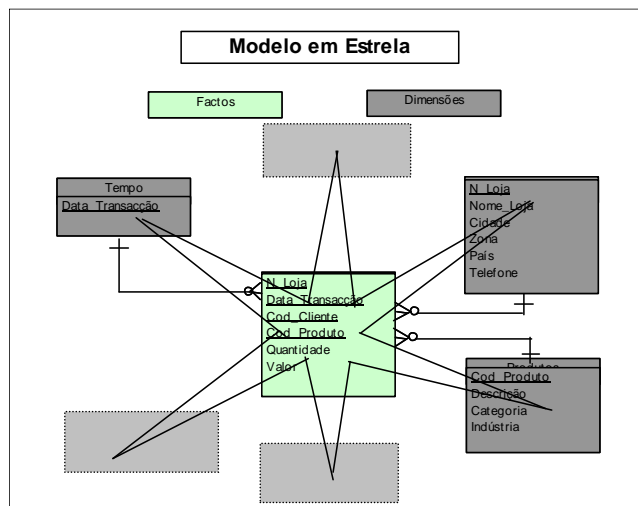
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela

Entre as tabelas de dimensões e as tabelas de factos, existe, habitualmente um **relacionamento de um para muitos (1:n)**

Entre uma tabela de dimensão podem estar associados várias ocorrências de factos (Existe existem exceções, que iremos analisar mais a frente)



Fonte Santos & Ramos (2017)

## 1.4. Projecto, implementação e administração de um DW

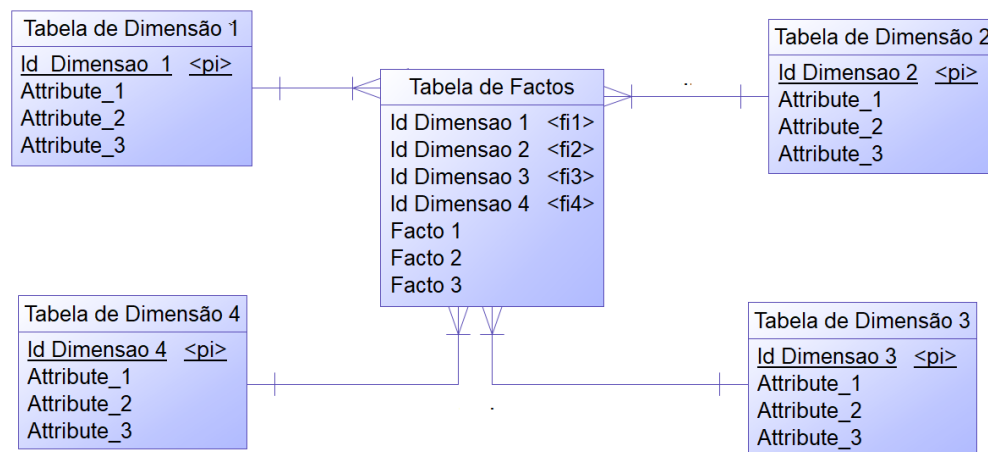
### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela

A tabela de factos corresponde ao assunto que se pretende analisar, normalmente uma componente do negócio: encomendas, vendas, compras, etc.

Esta tabela integra valores numéricos (por exemplo, valores de vendas em euros), que podem ser analisados utilizando diversas funções estatísticas.

Além dos atributos numéricos, a tabela factos integra também um conjunto de outros atributos, que constituem as chaves de ligação às tabelas de dimensão:



**Fonte Santos & Ramos (2017)**

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

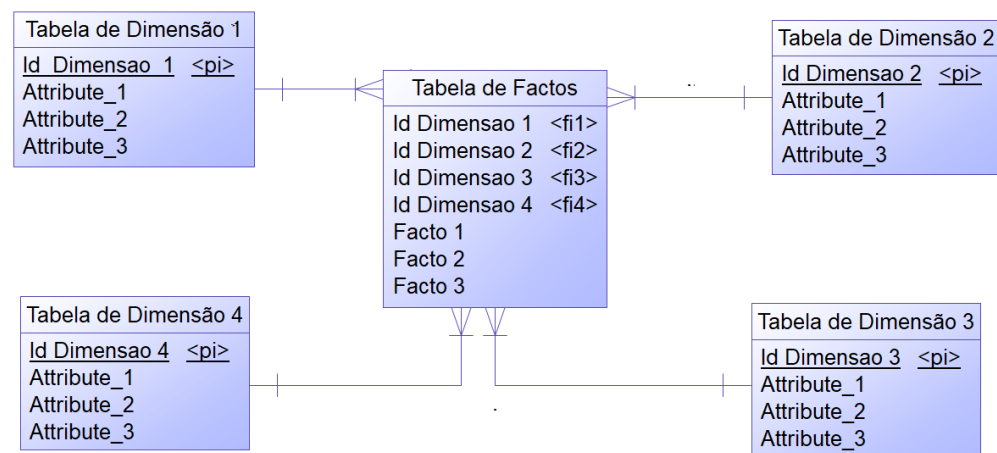
#### 1.5.1.5. Esquema em Estrela

As tabelas de dimensão possibilitam a análise de factos sob diferentes perspetivas, permitindo responder a diversas questões, tais como:

**“Quem?”**, **“Quando?”**, **“Onde?”** **“Porquê?”**, **“Como?”**, entre outras

Um esquema em estrela pode ter um qualquer número de dimensões

As tabelas de dimensão, são geralmente tabelas não normalizadas, podendo ter embebidas uma ou mais características



Fonte Santos & Ramos (2017)

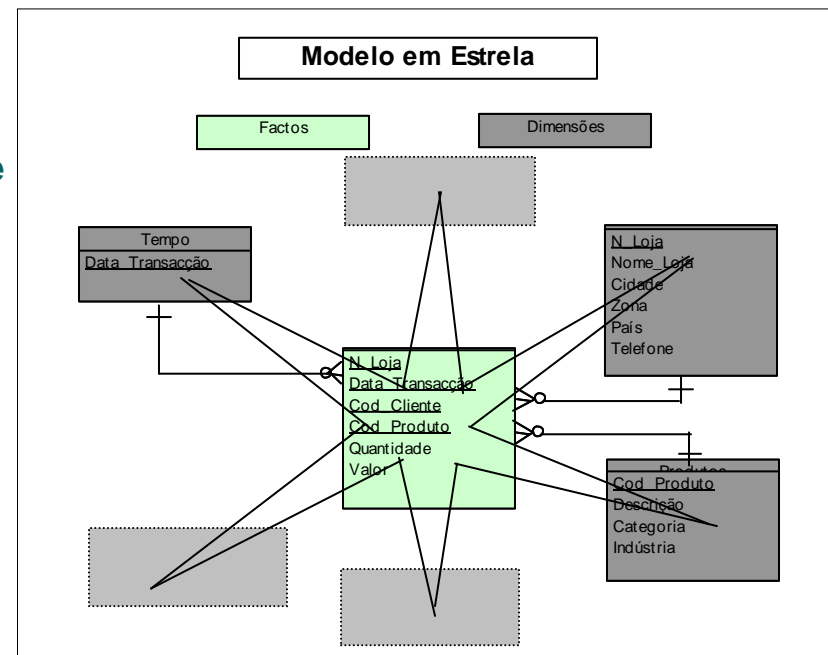
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela

Razão da seleção deste esquema:

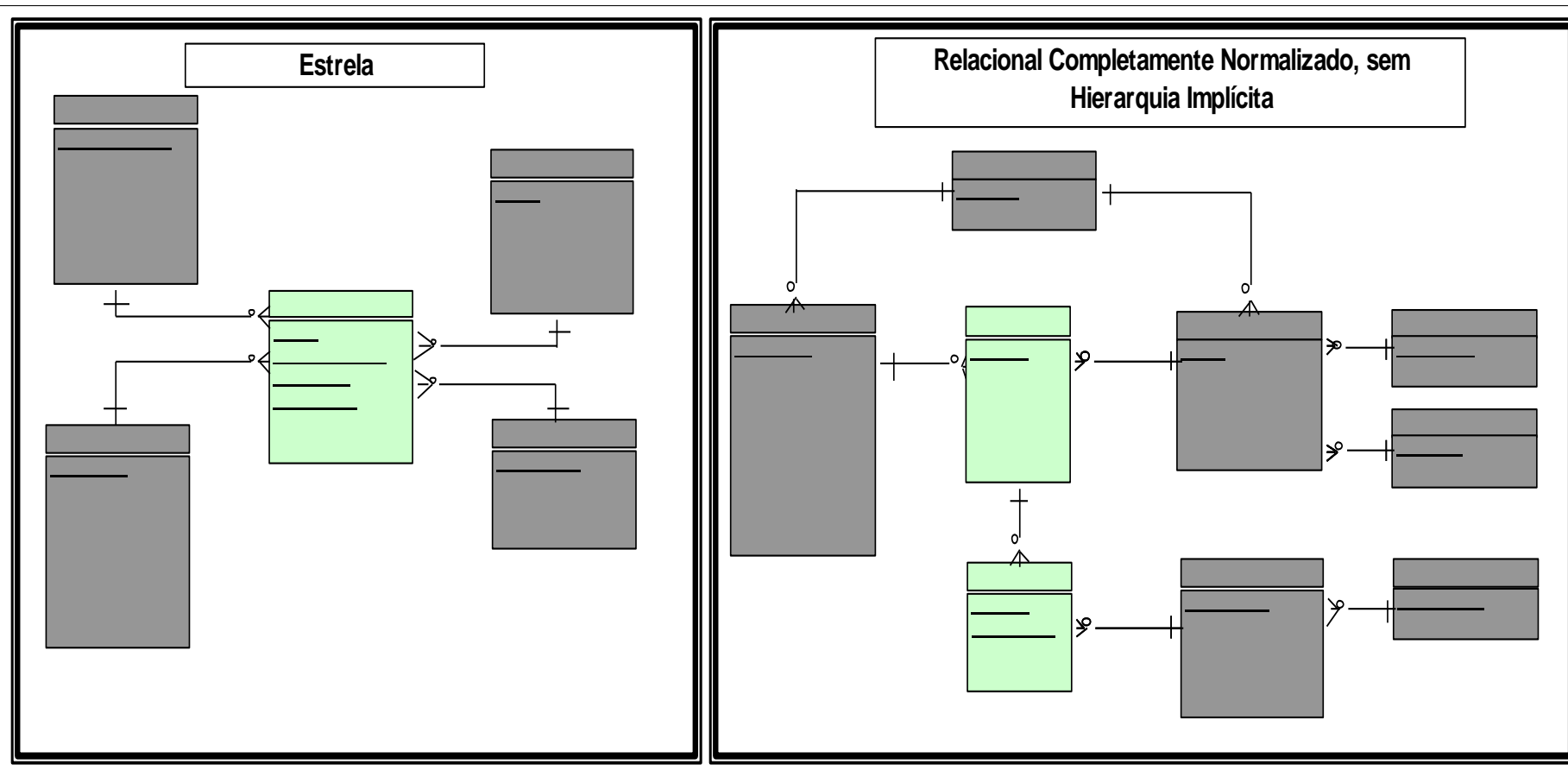
- **Distinção clara entre a natureza dos dados que constituem cada tipo de tabelas:**
  - Os factos são normal/ numéricos e em grande número (milhões ou biliões de linhas);
  - Do outro lado, nas tabelas dimensões, os dados são predominantemente alfanuméricos (características), em número muito inferior.
- **Evitam-se frequentes operações de junção;**
- **Permite que as operações de restrição de uma interrogação possam ser avaliadas só nas tabelas dimensão (a um custo reduzido, dadas as suas pequenas dimensões), só tocando a tabela factos, depois de encontrar o conjunto dos dados que passou as restrições.**



## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela



## 1.4. Projecto, implementação e administração de um DW

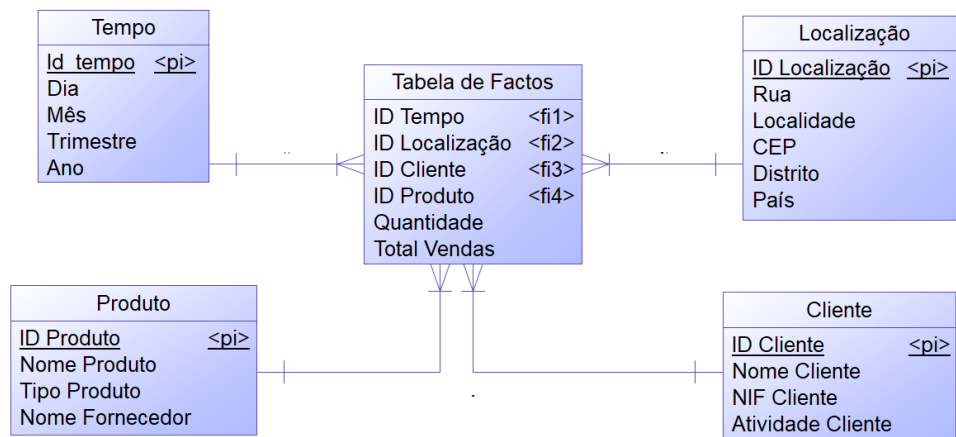
### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela - Exemplo

Pretendemos modelar um esquema em estrela para uma loja que deseja analisar as suas vendas ao longo de um período de tempo;

Desta forma identificamos as dimensões: **Tempo**, **Localização**, **Produto** e **Cliente**

Os factos a explorar são o **valor total de vendas e quantidades**.



As análises efetuadas aos factos permitem responder a perguntas que incluem as questões “Quando?”, “Onde?”, “O Quê?” e a “Quem?”

Fonte Santos & Ramos (2017)

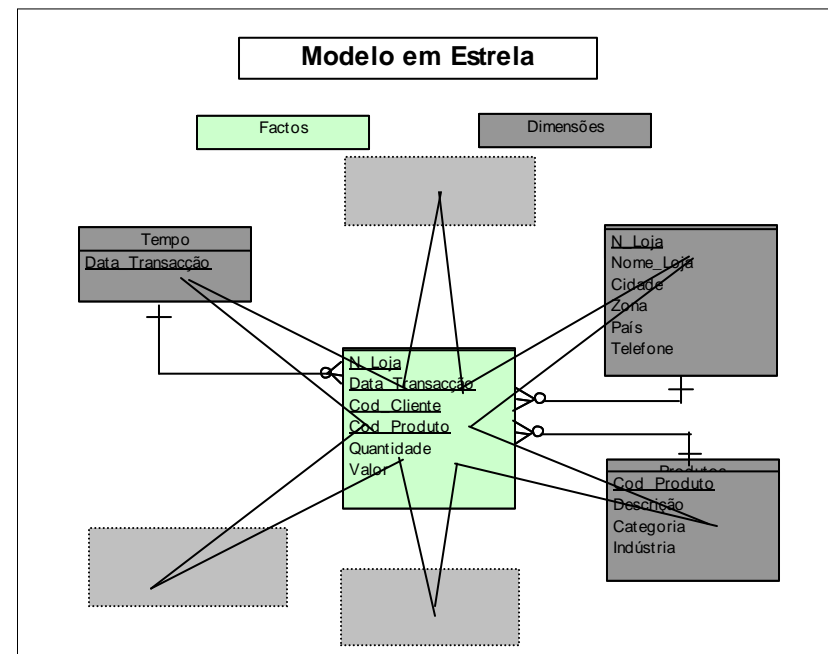
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1. Modelação Multidimensional

#### 1.5.1.5. Esquema em Estrela - Resumo

##### Resumo das características de um esquema em estrela clássico:

- Uma tabela factos única, com os dados de detalhe e sumarizados;
- A chave primária da tabela factos tem uma coluna relativa a cada dimensão;
- Cada dimensão é materializada por uma tabela desnormalizada.



## *1.5. Projecto, implementação e administração de um DW*

### *1.5.1. Modelação Multidimensional*

#### *1.5.1.6. Esquema em Floco-de-Neve*

#### *Esquema em Floco-de-Neve*

- Um esquema em floco de neve introduz um refinamento, adicionando as hierarquias de forma explícita, normalizando parcialmente ou completamente as tabelas dimensão.

#### **Razão da selecção deste esquema:**

- O esquema em estrela não permite suporte explícito para hierarquias de atributos, porque a hierarquia implícita é incluída como atributos nas tabelas dimensão (como cidade → zona ou produto → categoria).
- Já podemos saber que um determinado número de lojas pertence a uma determinada zona e que um conjunto de produtos é agrupado numa categoria.
- Permite também a criação das tabelas relativas às agregações, criando-se sucessivas tabelas factos, para cada uma das agregações a considerar.
- Evita que a informação redundante seja armazenada, uma vez que as dimensões estão devidamente normalizadas.



## *1.4. Projecto, implementação e administração de um DW*

### *1.5.1. Modelação Multidimensional*

#### *1.5.1.6. Esquema em Floco-de-Neve*

#### *Esquema em Floco-de-Neve*

##### Desvantagens:

- muito maior complexidade;
- a gestão mais complexa dos metadados;
- explosão do número de tabelas relativas às pré-agregações;
- segundo Kimball, há uma degradação de desempenho, em operações de browsing.

Aliás este autor só aconselha uma utilização marginal, como é o caso da existência de grandes blocos repetidos de dados demográficos, relativos à dimensão cliente e descrevendo a demografia da zona do código postal do cliente.

## 1.4. Projecto, implementação e administração de um DW

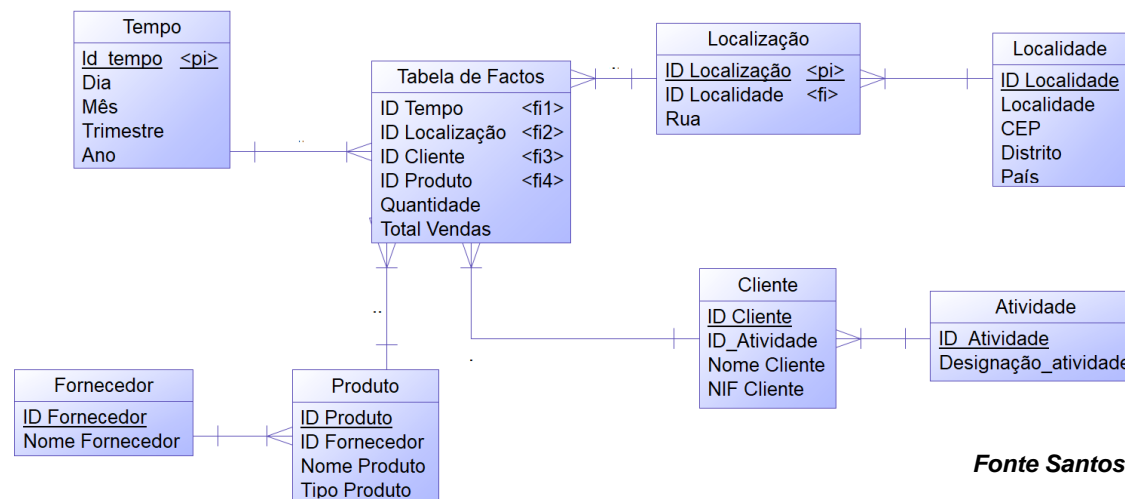
### 1.5.1. Modelação Multidimensional

#### 1.5.1.6. Esquema em Floco-de-Neve - Exemplo

Pretendemos modelar um esquema em estrela para uma loja que deseja analisar as suas vendas ao longo de um período de tempo;

Desta forma identificamos as dimensões: **Tempo**, **Localização**, **Produto** e **Cliente**

Os factos a explorar são o **valor total de vendas e quantidades**.



Fonte Santos & Ramos (2017)

A tabela Produto, Cliente e Localização foram normalizadas

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.1.7 Modelação Multidimensional

#### *1.5.1.7. Esquema em Constelação*

**Um esquema em constelação é um esquema que integra múltiplas tabelas de factos que partilham dimensões comuns**

**O esquema em constelação pode ser visto como um conjunto de esquemas em estrela, com dimensões que permitem a integração destes diversos esquemas.**

*Fonte Santos & Ramos (2017)*

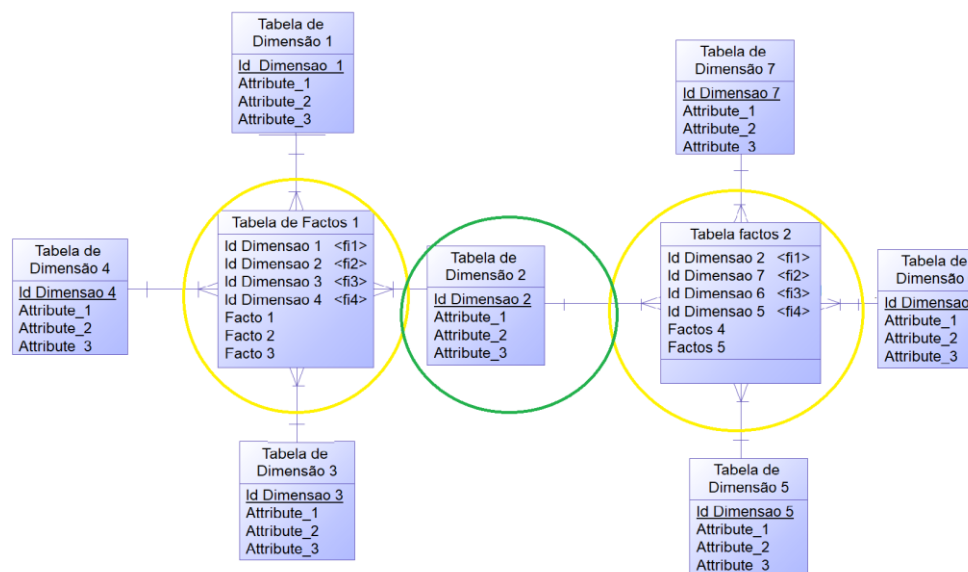
## 1.4. Projecto, implementação e administração de um DW

### 1.5.1.7 Modelação Multidimensional

#### 1.5.1.7. Esquema em Constelação

No caso dos esquemas em constelação, as diversas estrelas que os integram podem ser interligadas por mais do que uma dimensão

Neste caso a tabela dimensão 2 é partilhada pelo facto 2 e facto 1

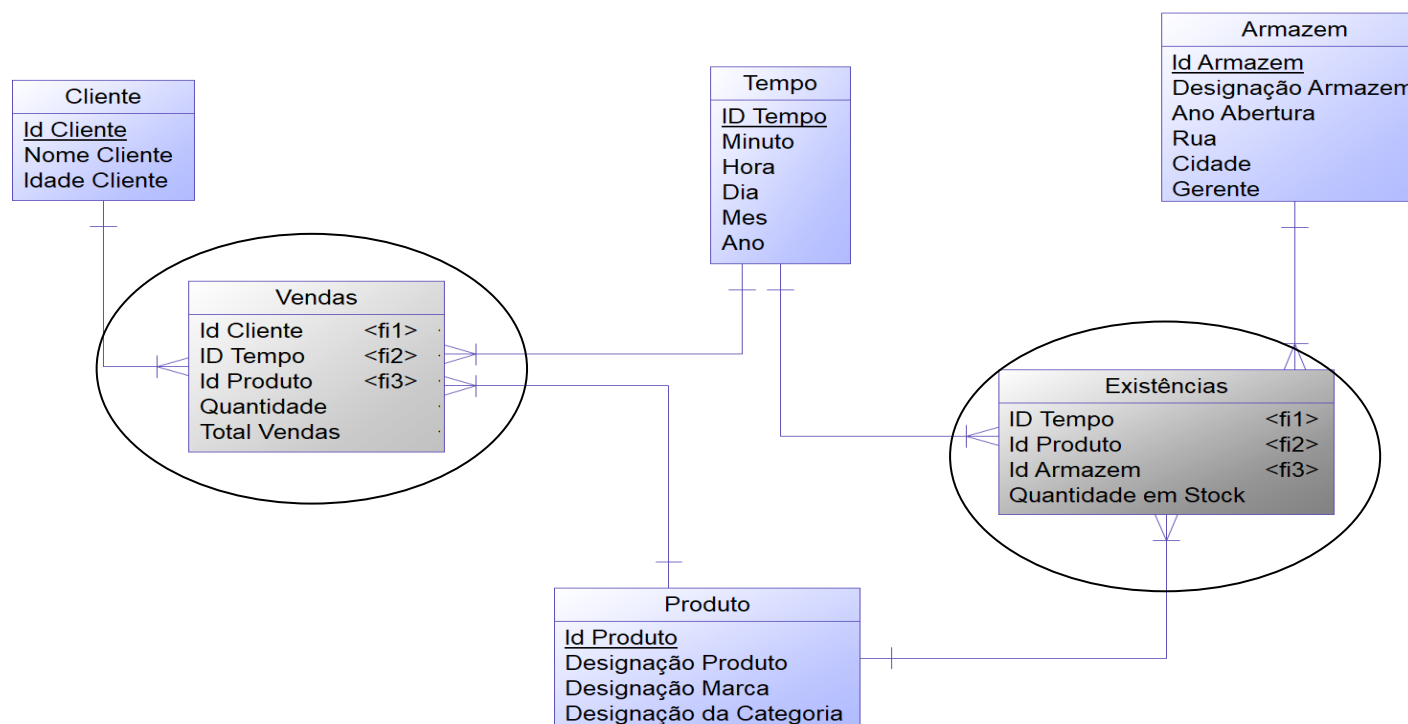


Fonte Santos & Ramos (2017)

## 1.4. Projecto, implementação e administração de um DW

### 1.5.1.7 Modelação Multidimensional

#### 1.5.1.7. Esquema em Constelação - Exemplo



Fonte Santos & Ramos (2017)

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.1. Modelação Multidimensional

#### *1.5.1.8. Modelação dum DW: Qual é o melhor esquema?*

## Qual é o melhor desenho?

- **Benchmarkings relativos ao desempenho podem ser utilizados para saber qual é o melhor esquema.**
  - **Esquema Floco-de-Neve:** Fácil de manter tabelas dimensão quando são muito grandes (reduz o espaço total);
  - **Esquema em Estrela:** Mais eficaz para browsing dos cubos de dados (menos operações de junção) – pode afectar o desempenho.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.2. Concepção de um DW: Processo em 4 fases

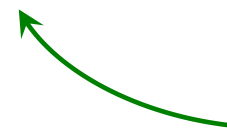
**Segundo Kimball**, “a criação de um DW dimensional é um processo de fazer corresponder as necessidades da comunidade dos utilizadores às realidades dos dados disponíveis”

**Requisitos de  
Negócio**



#### **Modelo Dimensional**

- 1. Processo de negócio**
- 2. Granularidade**
- 3. Dimensões**
- 4. Factos**



**Realidade dos  
Dados**

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW: Processo em 4 etapas*

#### *1.5.2.1. 1ª Etapa - Identificação dos Processos de Negócio*

##### Escolher o(s) processo(s) de negócio a modelar

- A identificação dos processos de negocio tem em consideração o levantamento de requisitos efetuados e que permite conhecer as necessidades do negócio e a priorização das mesmas, em termos de desenvolvimento.
- Conhecendo os processos a implementar, que dão origem às tabelas de factos, as próximas etapas do processo de modelação podem prosseguir
- Um processo de negócio é uma atividade realizada por uma organização, como por exemplo: efetuar vendas, efetuar encomendas, realizar exportações, registar estudantes



## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW: Processo em 4 etapas*

#### *1.5.2.2. 2ª Etapa - Identificação da Granularidade*

##### **Escolher a granularidade dos processos de negócio (nível atómico dos dados)**

- Este passo é de fundamental relevância, dado que, desta escolha, dependerá o grau de detalhe que será possível obter, numa consulta ao DW;
- Será o nível atómico dos dados que povoarão a tabela factos;
- Granularidades típicas que podem ser consideradas serão, por exemplo: transacções individuais, instantâneos (snapshots) individuais, diários ou mensais.
- A selecção da granularidade tem um impacto enorme no tamanho da base de dados, tendo de ser cautelosa qualquer aproximação que leve à selecção dos dados ao nível de maior detalhe;
- É determinante, para a selecção da granularidade, o tipo de consultas pretendidas;
- Da selecção da granularidade resultam, imediatamente, as dimensões primárias.
- Este detalhe influencia o conteúdo das tabelas de factos e das tabelas de dimensão

## 1.4. Projecto, implementação e administração de um DW

### 1.5.2. Concepção de um DW: Processo em 4 etapas

#### 1.5.2.2. 2ª Etapa - Identificação da Granularidade

Trata-se de um dos conceitos básicos do desenho dum DW

- A granularidade dos factos é importante:
  - O que quer dizer um simples facto?
  - **Nível de detalhe:**
    - Dado pela combinação dos níveis mais baixos em cada dimensão;
    - Exemplo: "vendas totais por loja, por dia e por produto".
  - Granularidade refere-se ao nível de detalhe ou sumarização mantida nas unidades de dados do DW;
- **Granularidade fina (baixa)** corresponderá a dados detalhados, e inversamente.
- Muitas vezes, a granularidade é a transacção do negócio
  - Exemplo: venda
  - Algumas vezes os dados são agregados (vendas totais por loja, por dia e por produto)
  - A agregação pode ser necessária devido a questões de escalabilidade
- De maneira geral, o detalhe a nível da transacção pode ser suportada
  - Excepto talvez imensos clickstreams, etc.

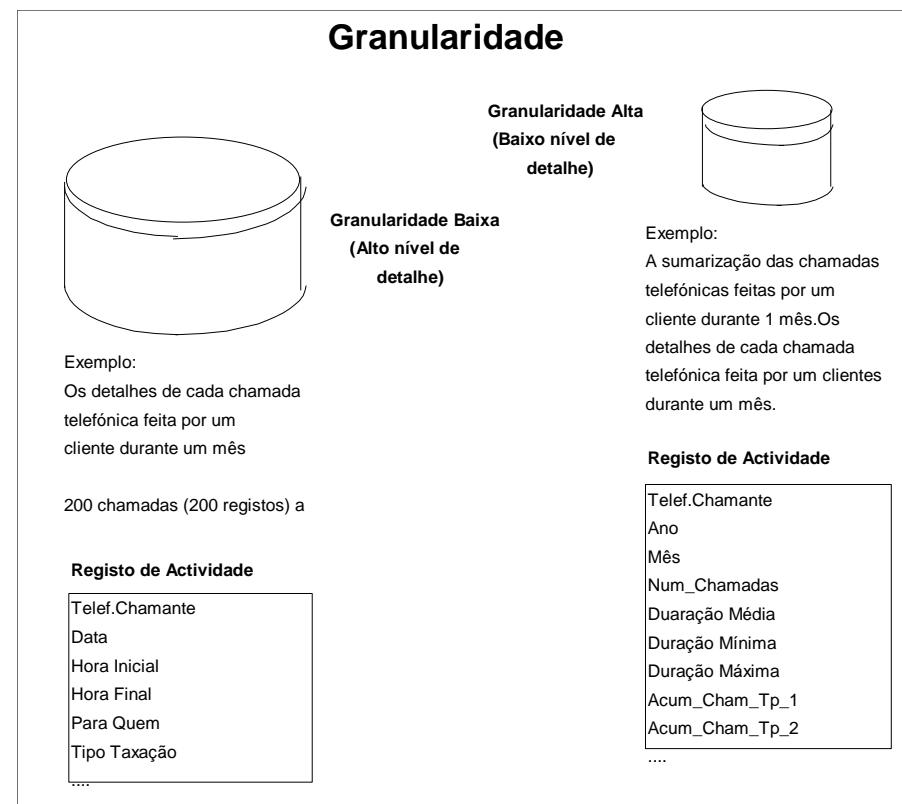
## 1.4. Projecto, implementação e administração de um DW

### 1.5.2. Concepção de um DW: Processo em 4 etapas

#### 1.5.2.2. 2ª Etapa - Identificação da Granularidade

Coloquemos agora duas questões ao DW e vejamos como responder:

- Quais foram as chamadas efectuadas pelo Sr. X, na semana passada?  
- resposta só pode ser dada pelo nível de granularidade baixo
- Em média, qual foi a duração das chamadas feitas, no ano anterior por utentes de Viseu?  
- resposta pode ser dada, utilizando o nível de granularidade alto ou baixo, claro que a custos diferentes, já que o I/O necessário para responder à questão, utilizando o nível de granularidade mais alto, seria muito menor.



Adaptado de: *Building the Data Warehouse*, W.H.Inmon

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW: Processo em 4 etapas*

#### *1.5.2.2. 2ª Etapa - Identificação da Granularidade (resumo)*

Quanto mais alto é o nível de granularidade:

- mais pequeno é o volume de dados;
- as consultas necessitam de aceder a poucos dados (menor I/O);
- não se pode responder a qualquer pergunta;
- acede-se a resultados baseados em estatísticas;
- é necessária uma máquina com menor desempenho.

Quanto menor é o nível de granularidade:

- maior é o volume de dados a tratar;
- as consultas necessitam de aceder a mais dados;
- pode responder-se a qualquer tipo de exigências;
- resultados baseados na realidade;
- é necessária uma máquina e SGBD capaz de aceder e tratar muitos dados (normalmente de processamento paralelo).

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW: Processo em 4 etapas*

#### *1.5.2.3. 3ª Etapa - Identificação das Dimensões*

### 3. Escolher as dimensões que caracterizarão cada registo da tabela factos

- Corresponde à identificação das dimensões que irão ser consideradas e que permitem analisar os processos de negócio sob diferentes perspetivas
- Cada dimensão será caracterizada por um conjunto de atributos (campos) que serão as colunas de cada tabela dimensão.
- Há que incluir o maior número possível de atributos, nas dimensões mais relevantes, porque será, através deles, que será possível efectuar restrições de uma consulta e, ainda,
- Porque são também eles que permitem as operações de drill-down / roll-up, mesmo na ausência de uma hierarquia explícita.
- Tempo, Loja, ...

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW: Processo em 4 etapas*

#### *1.5.2.4. 4ª Etapa – Identificação dos Factos*

#### Escolher as medidas que vão popular cada registo da tabela factos

- Os factos, métricas ou indicadores de negócio correspondem aos atributos numéricos que serão incluídos nas tabelas de factos e que detalham as medidas que permitem a análise dos diversos processos.
- Encontrar os Factos (medidas) que irão popular cada registo da tabela factos.
- São tipicamente valores numéricos aditivos, como quantidade vendida, valor de vendas, lucro bruto, que podem ser agregados, através de qualquer dimensão, ou um seu conjunto.
- Podem ser igualmente valores semi-aditivos, assim chamados, por não ser possível determinar, com exatidão, totais relativos a esse valor, segundo uma ou mais dimensões. Trata-se de valores especialmente ligados à dimensão tempo. A sua existência fica a dever-se ao nível de granularidade não ter sido a mais baixa possível e, desse modo, ter-se perdido algum detalhe.
- Como regra, pode dizer-se que todas as medidas que registem um nível estático, tais como, nível de stock, saldo financeiro de contas e medidas de intensidade como, por exemplo, temperaturas, são inerentemente não aditivas, na dimensão tempo [Kimball96].
- Ex. valor\_vendas\_actual, qnt\_vendas\_actual, valor\_custo, quantos\_clientes

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW*

#### *1.5.2.5. Concepção de um DW: Passos Adicionais*

5. Armazenar pré-cálculos na tabela factos
6. Completar as tabelas dimensão
7. Seleccionar a duração da base de dados
8. A necessidade de acompanhar as dimensões em mudança lenta
9. Decidir as prioridades nas interrogações e modos de interrogação

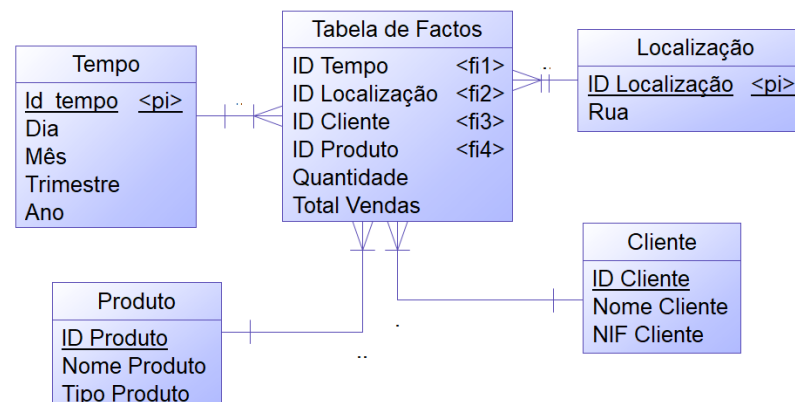
## 1.4. Projecto, implementação e administração de um DW

### 1.5.2. Concepção de um DW

#### 1.5.2.6. Calculo espaço para um DW ou DM

Calculo do tamanho da nossa datamart :

Será válido pensar que o numero de clientes não irá variar e que o numero de produtos não aumentará muito durante o tempo de vida da datamart.



Dimensão TEMPO, 365 dias X 12 meses X 10 anos = 43800 dias.

Dimensão CLIENTE, 50000 clientes.

Dimensão PRODUTOS, 10000 produtos.

Dimensão Localização, 5 armazéns.

Tamanho Médio de um Registo de Facto (**TMRF**), 6 campos \* 4 Bytes = 24 Bytes.

Percentagem de Espaço Ocupado pelas Dimensões em Relação aos Factos (**PEODRF**), 5%.

Tendo em conta que temos 6 campos por registo na tabela de factos obtemos a seguinte expressão:

- $(BD\_TEMPO * BD\_CLIENTE * BD\_PRODUTOS * BR\_LOCALIZAÇÃO * TMRF) * PEODRF$



## *1.4. Projecto, implementação e administração de um DW*

### *1.5.2. Concepção de um DW*

#### *1.5.2.6. Calculo espaço para um DW ou DM*

- Este valor aplicar-se-ia numa MOLAP em que teríamos todos os dados, nulos ou não, presente no sistema. No nosso caso, uma ROLAP com apenas a informação existente, temos que fazer a previsão de outro modo.
- Olhando para os dados presentes na tabela de factos, conseguimos concluir que o numero de vendas por ano ronda os 350000 registos. Alterando a expressão anterior para considerar este facto temos:
- Num. Médio Vendas Ano \* Tempo Vida Datamart \* PEODRF
- Percentagem de Espaço Ocupado pelas Dimensões em Relação aos Factos (**PEODRF**), 5%.

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Considerações de Concepção

#### *1.5.3.1. Forças e fraquezas do modelo multidimensional: Redundância*

- Só uma muito pequena redundância nas tabelas facto:
  - Múltiplas cópias de alguns dados podem existir na tabela factos (p.ex. dados do cabeçalho de uma encomenda são copiados para cada linha da encomenda)
  - Os dados relativos ao mesmo facto (geralmente) só são armazenados numa tabela factos
- A redundância é predominantemente devida às tabelas dimensão
  - As tabelas dimensão no esquema em estrela têm entradas redundantes para os níveis mais altos
- Problemas devidos à redundância?
  - Dados inconsistentes – o processo central de carga ajuda;
  - Tempo de actualização – o DW é optimizado para resposta a interrogações, não para actualizações;
  - Utilização de espaço: tabelas dimensão tipicamente ocupam menos de 5% do DW.
- Então: a redundância controlada é boa
  - Até um certo limite

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

#### 1.5.3.2. Relacionamento muitos-para-muitos

Num caso de um DW relativo à venda de carros, pode assumir-se que um carro pode ser vendido por um único vendedor a um único cliente. Mas pode não ser assim. Pode ser interessante saber dados demográficos de casais que comprem veículos. A resposta à questão é decerto importante: Que características demográficas dos casais afetam as vendas? Temos **um relacionamento do tipo muitos-para-muitos**. Como modelá-lo no DW?

- O método de captura de **relacionamentos muitos-para-muitos (M-N)** é idêntico ao usado no mundo transaccional. Entre a tabela factos e a tabela dimensão coloca-se uma tabela intercepção.
- Os registos da tabela intercepção são compostos pela chave da dimensão e normalmente um atributo da tabela factos. No caso das vendas de automóveis, teremos o n.º do contrato e o código do vendedor.

**Alternativa:** inclusão de múltiplos registos na tabela factos, permitindo, no caso em estudo, que um mesmo negócio tenha a intervenção de vários vendedores. Um novo registo será incluído na tabela factos para cada vendedor envolvido na venda. Há, no entanto, o perigo de erros, por contagem dupla de dados relativos a vendas, para medidas gerais (devendo cuidar-se a inclusão de uma cláusula UNIQUE).

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.3. Considerações de Concepção*

#### *1.5.3.3. Dimensões e Factos Degenerados*

#### **Dimensão Degenerada**

- Uma dimensão degenerada, não é mais do que uma dimensão fantasma, uma vez que os dados que, hipoteticamente, conteria, já estão guardados em outras dimensões, ficando assim, esvaziada. Um exemplo será, p. ex., o número de um talão de venda, o número de um contrato, etc.
- Em vez de criar a tabela dimensão adicional correspondente, podemos, simplesmente, armazenar o número da venda na tabela factos. Embora tecnicamente seja uma dimensão, foi armazenado como um atributo da venda – um facto.
- É útil quando se tenta associar os factos no DW com os documentos originais. No caso das vendas, permite, p.ex., a análise do cesto de compras, que, de outro modo, seria impossível.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

#### 1.5.3.3. Dimensões e Factos Degenerados

#### Factos Degenerados

- Tal como há dimensões degeneradas, há tb. **factos degenerados**. Resultam da tabela **intercepção correspondente a relações muitos-para-muitos**.
- Permite ao arquitecto do DW o registo de uma medida para cada instância dum relacionamento muitos-para-muitos.
- Voltando ao exemplo do DW de venda de automóveis, teremos a medida relativa às comissões recebidas. Neste caso o facto é degenerado, porque na tabela central teremos a comissão total e, na tabela intercepção, a comissão de cada vendedor.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

#### 1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”

Um facto é um facto: **ocorreu e é imutável!** O preço de uma venda ou o produto adquirido não vão ser alterados.

Mas as dimensões podem mudar!

- Os produtos alteram a sua descrição ou formulação;
- Os clientes, particularmente, estão em constante mudança: alteração de estado civil, de endereço, de número de filhos, etc.;
- Noutras dimensões, poderíamos encontrar também situações deste tipo (p. ex. na dimensão zonas de vendas, quando há uma redistribuição das zonas);
- Estamos perante uma situação que, apesar de ainda não abordada, não será, de forma alguma, marginal;
- Esta problemática foi denominada em [Kimball96] de “dimensões em alteração lenta”.
- Como pode o DW reflectir, com precisão, o ambiente em mudança, sem corromper a memória?

## 1.4. Projecto, implementação e administração de um DW

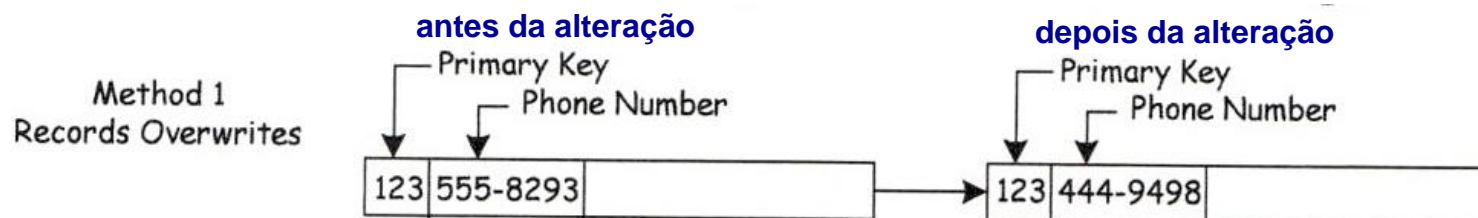
### 1.5.3. Considerações de Concepção

#### 1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”

**Cinco soluções:** há que decidir qual a mais adequada, tendo em conta a situação real.

**1. A mais directa:** simplesmente alterar o registo da dimensão;

- Quando a visão do passado e de futuro é alterada permanentemente
  - e.g. alterar as regiões de 3 para 4: as análises serão efectuadas em termos da nova estrutura geográfica.
- Será o mais apropriado quando a alteração ocorre em atributos não analíticos (e.g. número de telefone) ou quando se pretende corrigir erros;
- Mas... E se houver uma venda de um automóvel a uma senhora casada de meia idade e, há 17 anos, tiver havido outra venda ao mesmo cliente, que era, na altura, um rapaz solteiro?... (a legislação e a medicina moderna, faz destas coisas!);
- As alterações com efeitos retroactivos podem ter consequências perniciosas, levando a decisões erradas... Há uma corrupção da história!



Adaptado de: Object-Oriented Data Warehouse Design, William Giovino

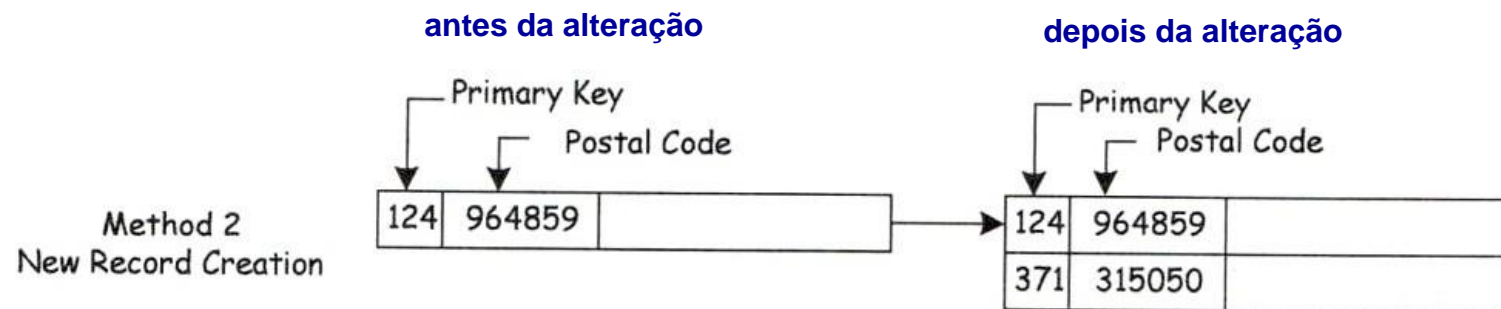
## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

#### 1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”

##### Cinco soluções:

2. **Em alternativa** podemos criar um novo registo na dimensão. Isto mantém a história passada, enquanto permite reflectir o estado actual do mercado.
- Nada de errado nos dados,
  - Mas... informação valiosa fica escondida! A própria alteração na dimensão pode ser valiosa, do ponto de vista analítico.



Adaptado de: *Object-Oriented Data Warehouse Design*, William Giovanazzo



## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

#### 1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”

##### Cinco soluções:

##### 3. Alterar a estrutura do registo na dimensão para conter o valor antigo e novo, incluindo a data de alteração.

- Assim, podemos reflectir, com precisão, o estado quando ocorreu o facto;
- Mas...
  1. Quais dos atributos serão de expectável alteração? Todos? Alguns? Como seleccioná-los?
  2. Quantas alterações serão expectáveis?



Adaptado de: *Object-Oriented Data Warehouse Design*, William Giovinazzo

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

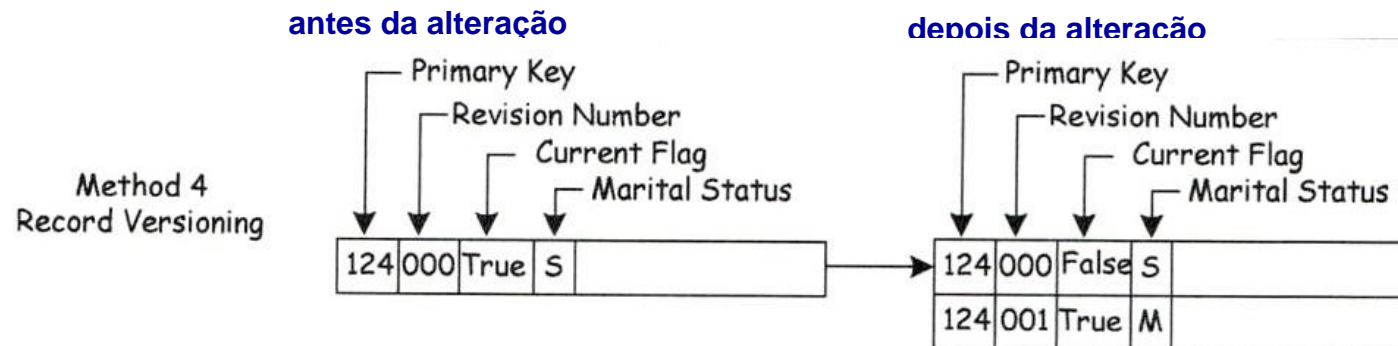
#### 1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”

##### Cinco soluções:

4. Modificar ligeiramente a solução 2. A criação de um novo registo na dimensão, quando ocorrer a alteração, inclui um número de versão, adicionado no final da chave primária.

- Os números de versão identificam a ordem na alteração do atributo;
- Incluir um campo binário para identificar qual o mais recente;
- Esta solução permite, não só capturar o estado corrente do objecto, mas acompanhar as mudanças – quando ocorreram e que efeito tiveram no comportamento do objecto.
- Mas... Esta solução complica a ligação entre as tabelas dimensão e a tabela factos. Em vez da junção se fazer através de um campo, passamos a ter uma chave concatenada, aumentando também o tamanho da chave da tabela factos...

Adaptado de: *Object-Oriented Data Warehouse Design*, William Giovinazzo



## 1.4. Projecto, implementação e administração de um DW

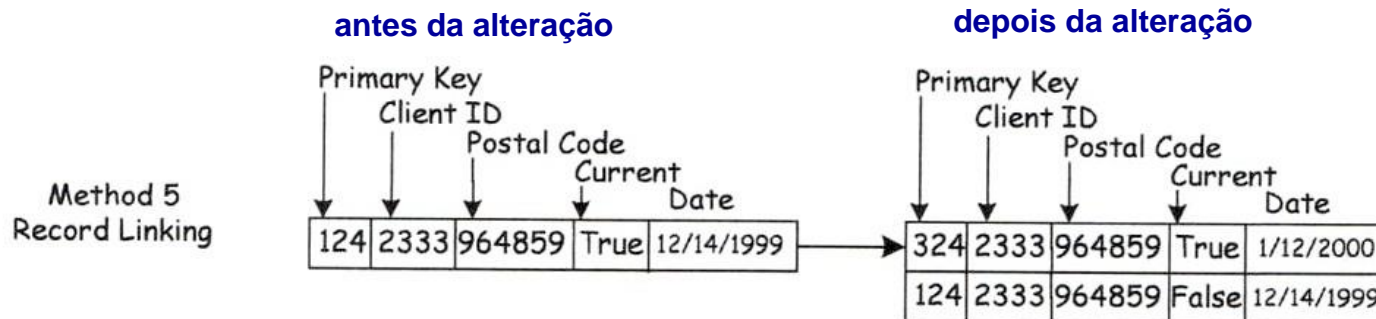
### 1.5.3. Considerações de Concepção

#### 1.5.3.4. Dimensões em Alteração Lenta “Slowly Changing Dimensions”

##### Cinco soluções:

5. Última solução. Combinar o método 2 e 4 de lidar com o problema. Quando uma dimensão se altera, um novo registo é criado com uma nova chave primária, tal como no segundo método. O ID do objecto (no sistema operacional) é também incluído, servindo para ligar as diferentes versões. Uma flag é incluída para indicar qual a versão do registo dimensão mais recente.
- Desta forma, retém-se a história dos objectos descritos pela dimensão e não tem o problema indicado na solução 4; parece ser a solução óptima.
  - Mas... a solução nem sempre está disponível. Pode ser complicada de gerir, se os dados provierem de fontes díspares (sobrepostas ou inconsistentes): não se pode contar com o ID dos sist. operacionais para identificar registos.

Adaptado de: *Object-Oriented Data Warehouse Design*, William Giovinazzo



## *1.4. Projecto, implementação e administração de um DW*

### *1.5.3. Considerações de Concepção*

#### *1.5.3.5. Dificuldades na Implementação de um DW*

- O tempo de concepção, desenvolvimento e instalação é grande para a criação de um DW;
- Potencialmente, pode levar anos a criar e a manter eficazmente um Data Warehouse;
- A qualidade e consistência dos dados são um assunto da maior relevância;
- Rever as projecções de utilização regular, para satisfação dos requisitos actuais;
- O Data Warehouse deve ser concebido de forma a acomodar a adição e a exaustão de fontes de dados, sem carecer de grande redesenho;
- A administração de um Data Warehouse irá requerer muito maiores competências do que quando se trata de bases de dados tradicionais.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Considerações de Concepção

#### 1.5.3.6. Mitos relativos à Modelação Dimensional

- **Mito 1: Modelos Dimensionais e Data Marts são só para dados sumarizados**
  - A inexistência de dados a nível de detalhe adequado impede a resposta a interrogações não previsíveis do analista (ad hoc queries).
- **Mito 2: Modelos Dimensionais e Data Marts são departamentais, não soluções empresariais**
  - Os DMarts devem ser organizados à volta de processos de negócio (encomendas, facturas, serviços de assistência) e não desenhados limitados pelas fronteiras de um departamento. Múltiplas funções de negócio analisam, muitas vezes, as mesmas métricas que resultam de um único processo de negócio. Evita-se assim duplicação das medidas nucleares por múltiplas bases de dados.
- **Mito 3: Modelos Dimensionais e Data Marts não são escaláveis**
  - As tabelas facto podem conter dezenas, centenas de milhares de milhões de linhas. Os fabricantes de SGBDR incorporaram capacidades para otimizar a escalaridade e desempenho atendendo aos modelos dimensionais.

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Considerações de Concepção

#### *1.5.3.6. Mitos relativos à Modelação Dimensional*

- **Mito 4: Modelos Dimensionais e Data Marts só são apropriados quando existe um padrão de utilização previsível**
  - A simetria do modelo dimensional torna as estruturas de dados extremamente flexíveis e adaptáveis às alterações. O segredo da flexibilidade a alterações de padrão de interrogações é a criação de tabelas facto com granularidade mais fina.
- **Mito 5: Modelos Dimensionais e Data Marts não podem ser integrados e não levam a soluções de excelência**
  - Podem ser integrados se estiverem conformes com a arquitectura de bus data warehouse.

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Considerações de Concepção

#### *1.5.3.7. Ciladas a Evitar*

1. Enamorar-se da tecnologia e dados em vez de focar nos requisitos e objectivos do negócio;
2. Falhar no envolvimento ou recrutamento de um visionário de gestão influente, acessível e razoável como o patrocinador do data warehouse;
3. Embarcar num projecto galáctico, multianual, ainda que apelativo, em vez de prosseguir com um projecto mais gerível, com desenvolvimento iterativo;
4. Alocar energia para construir uma estrutura de dados devidamente pensada e depois ultrapassar o orçamento antes de construir uma área de apresentação viável baseada em modelos dimensionais;
5. Dar mais atenção às questões de desempenho da área de armazenamento e de extracção e à facilidade de desenvolvimento do que ao desempenho de resposta às interrogações e facilidade de uso;

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Considerações de Concepção

#### *1.5.3.7. Ciladas a Evitar*

6. Tornar os dados, supostamente interrogáveis na área de apresentação, demasiado complexos. Implica um suporte acrescido aos utilizadores que apreciarão soluções mais simples;
7. Popular os modelos dimensionais numa base isolada sem olhar à arquitectura de dados que os ligue, utilizando dimensões partilháveis e conformes;
8. Carregar só dados sumarizados nas estruturas dimensionais da área de apresentação;
9. Presumir que o negócio, os seus requisitos e análises, e os dados subjacentes e as tecnologias de suporte são estáticos;
10. Negligenciar o saber que o sucesso de um DW está ligado directamente à sua aceitação por parte dos utilizadores. Se os utilizadores não aceitaram o DW, como a base de melhor tomada de decisão, então os esforços terão sido um exercício de futilidade.



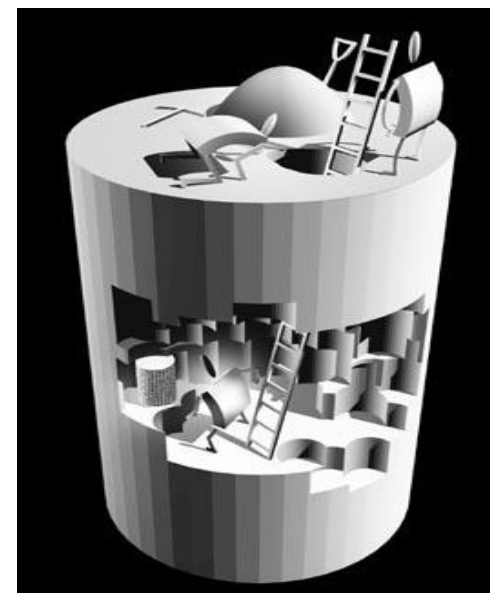
## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Implementação de um Data Warehouse

#### 1.5.3.8. Considerações Técnicas

Incluem um conjunto de questões técnicas:

- A plataforma hardware que suportará o DW;
  - TPC-H benchmarks (<http://www.tpc.org/default.asp>);
  - Processamento paralelo? Perigos? (muitas vezes um processo tem componentes paralelas e outras série).
- O SGBD que suporta a base de dados do DW;
  - Grande tamanho das bases de dados;
  - Necessidade de processar interrogações complexas e ad hoc num tempo curto;
  - Vários graus de paralelismo disponíveis nos SGBDs, mas há requisitos especiais para DW que devem ser suportados.
- Índices disponíveis e a utilizar;
  - Índices bitmap;
  - Índices junção e junção-estrela.



## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Implementação de um Data Warehouse

#### *1.5.3.8. Considerações Técnicas*

Incluem um conjunto de questões técnicas:

- A infraestrutura de comunicação que liga o DW, DMarts, sistemas operacionais e utilizadores
  - O volume de dados a movimentar pode ser bastante significativo – é necessária uma largura de banda, implicando, porventura, expansão da rede.
- A plataforma hardware e software que suporta o repositório de metadados;
- O sistema que permite a gestão centralizada e administração de todo o ambiente.

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Implementação de um Data Warehouse

#### *1.5.3.9. Considerações de Implementação*

Um DW não pode ser, simplesmente adquirido e instalado – a sua implementação requiere a integração de muitos produtos na arquitectura

- Para se ver a complexidade, há que pensar os passos na implementação dum DW:
  - Recolher e analisar os requisitos de negócio;
  - Criar o modelo de dados e desenho físico para o DW;
  - Definir as fontes de dados;
  - Escolher a tecnologia da base de dados e a plataforma de hardware;
  - Extrair os dados das bases de dados operacionais, transformá-los, purificá-los e carregá-los no DW;
  - Escolher as ferramentas de acesso e reporting;
  - Escolher o software de análise de dados e apresentação;
  - Actualizar (refrescar) o DW;
  - Optimizar o DW.

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.3. Implementação de um Data Warehouse*

#### *1.5.3.10. Ferramentas e Utilitários Back-End*

- **Extracção de dados**
  - Obtêm dados de fontes múltiplas, heterogéneas e externas
- **Purificação de dados:**
  - Detectam erros nos dados e rectificam-nos, quando possível
- **Transformação de dados:**
  - Convertem dados de formato legacy ou host para o formato warehouse
- **Carga:**
  - Ordenam, sumarizam, consolidam, calculam vistas, verificam a integridade, e criam índices e partições
- **Refrescar**
  - Propagam as actualizações das fontes de dados para o warehouse

## *1.4. Projecto, implementação e administração de um DW*

### *1.5.3. Implementação de um Data Warehouse*

#### *1.5.3.11. Soluções integradas*

- A maioria das soluções propostas por fabricantes de produtos para DWing consistem numa base de dados relacional para o repositório DW, software de gestão do DW, e ferramentas de acesso e reporting.
- Uma pesquisa nos sites dos grandes fabricantes de produtos de BD e de ferramentas OLAP e reporting, permitirá conhecer as soluções propostas e efetuar uma análise comparativa.

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Implementação de um Data Warehouse

#### 1.5.3.11. Soluções integradas

- **Produtos ORACLE**
  - Oracle Database 12c (<http://docs.oracle.com/database/121/index.htm>)
    - Oracle Hexadata (<http://www.oracle.com/technetwork/database/exadata/exadata-technical-whitepaper-134575.pdf>)
    - Database inMemory
    - Partitioning
    - Big Data appliance
    - Analytical SQL (<http://www.oracle.com/technetwork/database/bi-datawarehousing/sql-analytics-index-1984365.html>)
    - Advanced Analytics (<https://www.oracle.com/database/advanced-analytics/index.html>)
    - Query Optimization
    - OLAP (<http://www.oracle.com/technetwork/database/options/olap/index.html>)
  - Link com informação: <https://docs.oracle.com/database/121/DWHSG/>

## 1.4. Projecto, implementação e administração de um DW

### 1.5.3. Implementação de um Data Warehouse

#### 1.5.3.11. Soluções integradas

- **Microsoft SQL Server**
  - Microsoft SQL Server 2012/2014 (<http://www.microsoft.com/en-us/server-cloud/products/sql-server/>)
  - SQL Server 2012/14 Business Intelligence Development Studio (<http://sqlmag.com/sql-server-2014/q-where-business-intelligence-development-studio-bids-sql-server-2014>)
  - Report Server
  - SQL Server Integration Services (SSIS) para implementar as operações ETL que são necessárias para construir o DW. Inclui:
    - Ferramenta ETL
    - Conectividade a fontes de dados não SQL Server, e.g. Oracle, Teradata, DB2 mainframe, and ERP Systems
  - Analysis Services (há um tutorial disponível, bastando pesquisar por SQL Server 2012/14 Analysis services)

## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Implementação de um Data Warehouse

#### *1.5.3.11. Soluções integradas*

- Teradata
  - Data Warehousing & Analytics
  - Products
  - Integrated Data Warehousing
  - Advanced Analytics
  - Data Lake Products
  - Cloud for Analytics
  - Big Data Architecture Software
  - Hadoop Products / Open Source

Link: <http://www.teradata.com/about-us/>



## *1.4. Projecto, implementação e administração de um DW*

### 1.5.3. Implementação de um Data Warehouse

#### *1.5.3.11. Soluções integradas*

- IBM - Data Warehousing & Analytics (<http://www.ibm.com/analytics/us/en/technology/data-warehousing/>)
- DB2 (<http://www.ibm.com/analytics/us/en/technology/db2/>)
- Business intelligence
- IBM InfoSphere Warehouse
- Predictive analytics
- IBM Cognos software (<http://www-01.ibm.com/software/analytics/cognos/>)
  - IBM Cognos Analytics; IBM Cognos TM1®; IBM Watson™ Analytics; IBM Cognos Express®
- RedBrick Warehouse (<http://www-03.ibm.com/software/products/en/redbrick>)
- IBM WebSphere DataStage (<http://www-03.ibm.com/software/products/en/ibminfodata>)
- DB2 Query Management Facility
- Informix Tools (<https://www-01.ibm.com/software/data/informix/tools/>)
- ...

# QUESTÕES

Filipe Sá

Filipe.sa@isec.pt