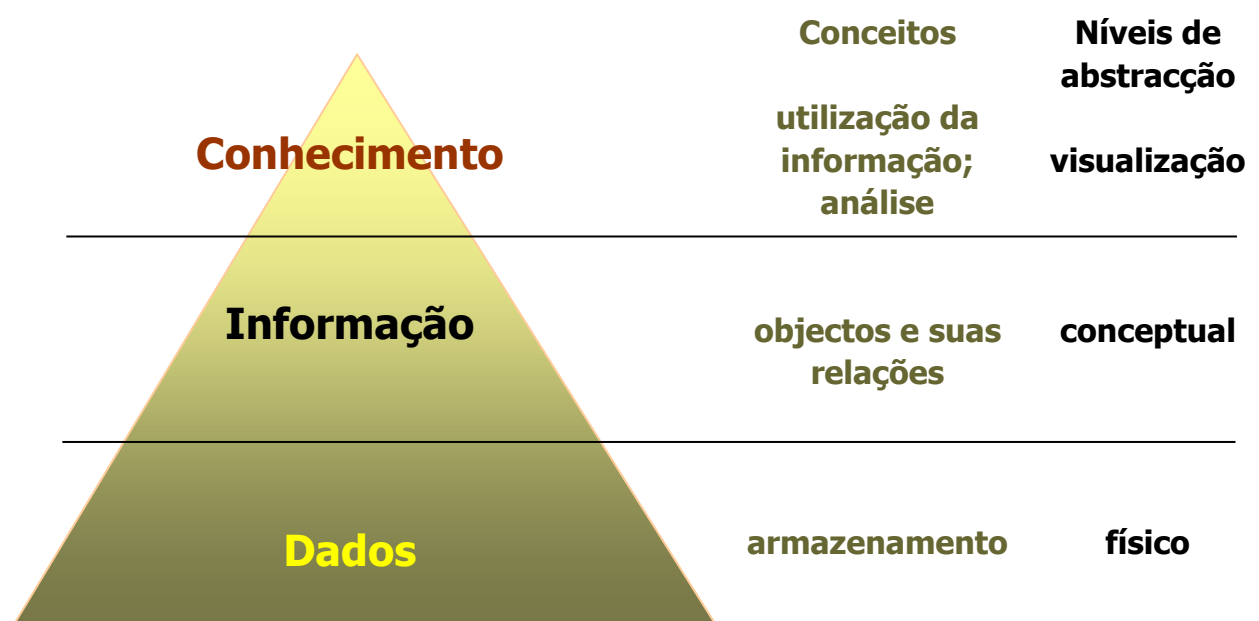


# OBJECTIVOS DOS PROJECTOS DE ESTRUTURAÇÃO DE DADOS

- ▶ ADEQUAÇÃO AOS OBJETOS
- ▶ INTEGRIDADE/ RIGOR
- ▶ DISPONIBILIDADE
- ▶ FACILIDADE DE ACTUALIZAÇÃO E PESQUISA
- ▶ ARMAZENAMENTO EFICIENTE

## OS TRÊS NÍVEIS DE INFORMAÇÃO



## OS TRÊS VÉRTICES DA GESTÃO DE INFORMAÇÃO

- Análise funcional da informação
- Desenvolvimento da aplicação que gere a aplicação
- Representação da informação

# CONCEITO DE BASE DE DADOS

Colecção de Dados que representam um negócio

Entidades (Produtos, Armazéns, Clientes, ...)

Relações entre entidades (a venda de produtos a um cliente)

SGBD - Sistema de Gestão de Base de Dados

Programa para suportar a gestão e utilização dos dados

Visão geral de assuntos no universo das bases de dados



## **Desenho/ Projeto de Base de Dados**

- Como descrever um negócio em termos da informação armazenada num SGBD



## **Análise de Dados**

- Como responder a questões sobre o negócio através da pesquisa da informação sobre a base de dados



## **Concorrência e Robustez**

- Como permite o SGBD o acesso concorrente a vários utilizadores?
- Como protege a informação das falhas do sistema?



## **Eficiência e escalabilidade**

- Como o SGBD consegue armazenar muita informação e responder às pesquisas de forma eficiente?

## PERSPECTIVA HISTÓRICA

### ⌘ primeiro SGBD de aplicação genérica

- IDS - Integrated Data Store (GE 1960s)
- usava um modelo de dados em rede

### ⌘ primeiro SGBD desenvolvido pela IBM para servir de base a aplicações de gestão de informação IBM

- IMS - Information Management System (IBM, fins anos 1960)
- usava um modelo de dados hierárquico

### ⌘ criação do modelo de dados actualmente mais usado pelos SGBD

- modelo de dados relacional (Edgar Codd - IBM research 1970)
- provocou o desenvolvimento de vários SGBD baseados nesse modelo
- iniciou a comercialização em grande escala de SGBD e suas aplicações
- os SGBD comerciais tornaram-se prática corrente da gestão de informação nas empresas

### ⌘ Consolidação dos SGBD e aparecimento do standard SQL (década de 1980)

- os SGBD passam a ser usados universalmente
- linguagem de pesquisa universal: SQL - Structured Query Language, para bases de dados relacionais (IBM system R)
- standardização do SQL (fins 1980)
- desenvolvimento de transacções e concorrência num SGBD (James Gray)

### ⌘ Avanços em várias áreas dos SGBD (década 1990, início sec XXI)

- linguagens de pesquisa mais poderosas
- modelos de dados mais ricos (objecto, *object-relational*)
- suporte de análise complexa de dados de cada parte da empresa

- imagens e texto em SGBD
- Datawarehouses e análise de dados especializada

### ⌘ Pacotes com funcionalidades orientada: ERP - Enterprise resource planning

- Baan, SAP, Navision, ...
- oferecem suporte a **processos típicos** de organizações e empresas
- armazenam os dados em SGBD relacionais
- adaptam o funcionamento a diferentes empresas - **parametrização**
- diminuem o custo em relação a um desenvolvimento de raiz

### ⌘ Entrada dos SGBD na era da internet e sites dinâmicos

- A informação Web deixa de estar armazenada em ficheiros e passa a estar armazenada num SGBD usando-se o SQL para pesquisas e transforma-se o resultado em HTML/ XML

## ARMAZENAMENTO - DOIS MODOS

- FICHEIROS INDIVIDUAIS
- BASE DE DADOS

Sistema Controlado de Informação

### Características do modo de armazenamento

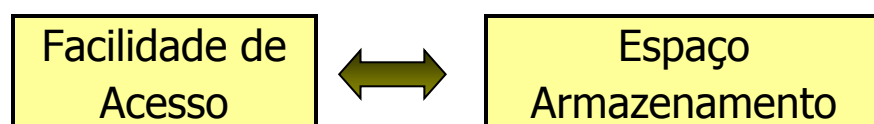
#### ➤ FICHEIROS:

- Rapidez do projecto e construção
- Conceitos mínimos de disponibilidade e segurança
- Dificuldade de evolução
- Alterações no armazenamento obrigam a alterações no código
- Obriga a REDUNDÂNCIA

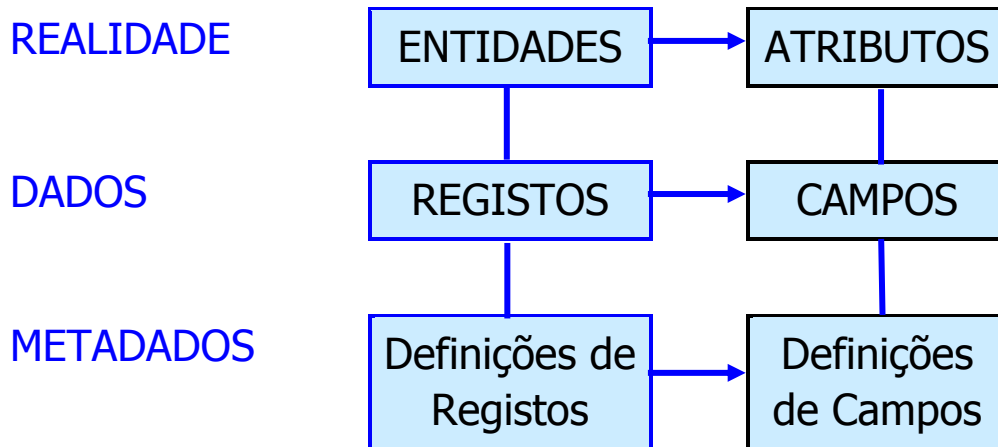


#### ➤ BASE DE DADOS:

- Motor de Base de Dados - SGBD, capaz de:
  - Criar
  - Actualizar
  - Modificar
  - Disponibilizar
  - Partilhar
  - Manter consistência
- Facilidade de evolução
- Permite criar vistas de dados (Views) sem preocupações com a estrutura
- ⚡ ▪ Todos os dados juntos ➡ Vulnerabilidade
- ⚡ ▪ Dificuldade balancear:

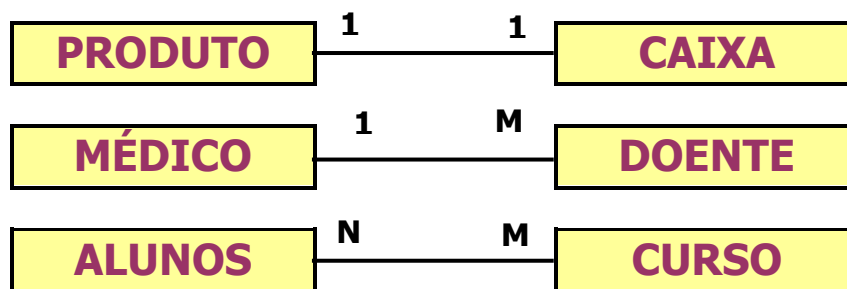


## CONCEITOS DE DADOS



**ENTIDADES** → OBJECTOS

**RELAÇÕES** → ASSOCIAÇÕES ENTRE OBJECTOS  
(perspectiva tradicional)



As relações 1-1 e N-M correspondem a visões incompletas e imperfeitas do Negócio e terão de ser sempre exploradas, percebidas e transformadas em relações 1-N.

## ATRIBUTOS



## REGISTOS

↓  
Colecção de Campos  
Ocorrências de Entidades

## CHAVES

Campos que identificam o registo

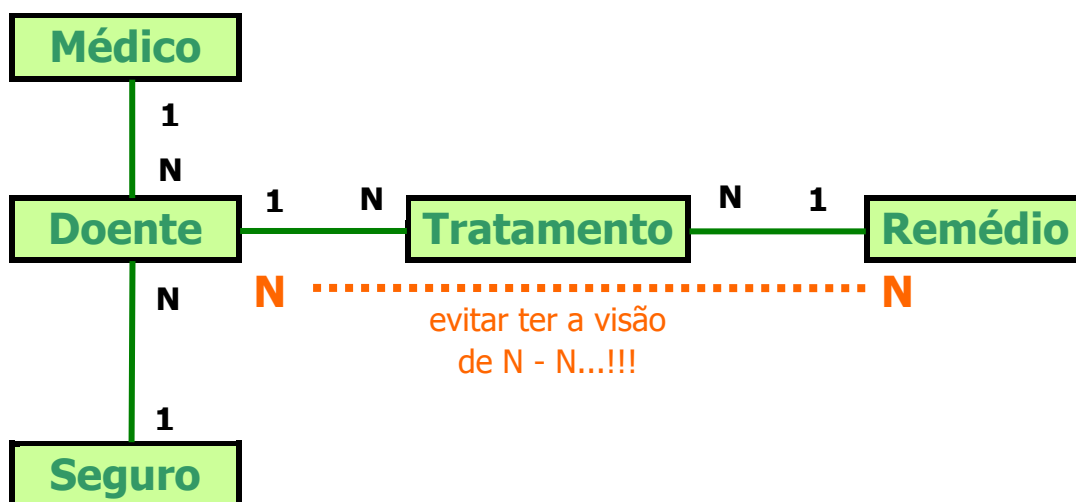
Primária: Um registo

Secundárias: Colecção de Registos

## METADADOS

Informação que descreve campos, registos e suas relações

## DIAGRAMA: ENTIDADE - RELAÇÃO



# ORGANIZAÇÃO

## ❑ **FICHEIROS** - Colecções de Registos

**TIPO** (quanto aos dados):

<b>MASTER</b>	Registos estáveis Campos muito alteráveis
<b>TABELAS</b>	Registos e campos estáveis
<b>TRANSACÇÕES</b>	Alteram os Masters
<b>TRABALHO</b>	Temporários
<b>IMPRESSÃO</b>	

**ORGANIZAÇÃO:**

<b>Sequencial</b>	Registos com ordem física
<b>Directo</b>	Registos com ordem lógica (encadeados)
<b>Hashed</b>	Acesso directo ao registo a partir da chave
<b>Indexados</b>	Chaves em ficheiros separados
<b>Sequencial - Indexada</b>	Registos em blocos por ordem física. Os blocos têm índice



## ❑ **BASE DE DADOS**

MODELO LÓGICO DE DADOS



DIFERENTES FORMAS DE VER OS DADOS

MODELO FÍSICO

COMO ARMAZENAR/ RELACIONAR/ ACEDER

ESTRUTURA: 3 FORMAS PRINCIPAIS

### ❖ **HIERARQUICO (ÁRVORE)**

- Cada entidade não tem mais que uma entidade ascendente
- Só comporta relações 1:M ou 1:1

### ❖ **REDE**

- Cada entidade pode ter qualquer número de subordinados ou superiores
- As entidades ligam-se por cadeias

### ❖ **RELACIONAL**

- Consiste numa ou mais tabelas bidimensionais (**Relações**) onde as linhas são os registos e as colunas os atributos

**Em que difere da folha de cálculo ?**

**DIFERE:**

- Volume de informação
- Permite ligar duas ou mais tabelas que aparecem como uma única
- Minimiza REDUNDÂNCIA

## MODELO HIERARQUICO

- \* GRANDE DEPENDÊNCIA ENTRE A DESCRIÇÃO DA ESTRUTURA DE DADOS E A MANEIRA COMO ESTÃO REGISTRADOS

- \* EXEMPLO: STOCKS

ASSOCIAÇÃO: **FORNECEDORES**  
**PRODUTOS**



F_COD	V_PUNIT	P_COD
F_NOME	V_QMIN	P_NOME
F_MOR	V_ATRASO	P_COR
		P_PESO

### CONTEÚDO DA BASE

F_COD	F_NOME	F_MOR					
F1	RUI	PORTO					
{			P_COD	P_NOME	P_COR	P_PESO	V_PUNIT
			P1	PREGO	AZUL	12	3
			P2	PARAFUSO	VERM	17	4
			P3	PORCA	VERDE	17	5
			P4	PERNO	AMARELO	14	2
			P5	CARRETO	AZUL	15	7
{	F2	EVA	BRAGA				
			P1	PREGO	AZUL	12	3
			P3	PORCA	VERDE	17	6

{ Segmento Fornecedores

{ Segmento Produtos

CADA SEGMENTO PRODUTO ESTÁ ASSOCIADO A UM OU MAIS SEGMENTOS FORNECEDOR

### UNIDADE DE ACESSO: SEGMENTO

- ➡ NÃO SE PODE ACEDER A UM SEGMENTO SEM ACEDER PRIMEIRO AO SEU SUPERIOR HIERÁRQUICO
- ➡ PARA LISTAR OS PRODUTOS É NECESSÁRIO ANALISAR PARA CADA FORNECEDOR A SUA LISTA DE PRODUTOS
- ➡ NÃO HÁ UMA FORMULAÇÃO NATURAL DAS PERGUNTAS PARA OBTER OS DADOS DA BASE

### QUESTÕES CURIOSAS

- ⦿ PARA INSERIR UM PRODUTO PARA O QUAL NÃO SE CONHECE O FORNECEDOR HÁ QUE CRIAR UM FICTÍCIO.
- ⦿ SE SUPRIMIR UM FORNECEDOR TODOS OS SEUS PRODUTOS DESAPARECEM, MESMO OS QUE SÃO ÚNICOS.
- ⦿ A ACTUALIZAÇÃO DE UM PRODUTO OBRIGA A PERCORRER TODOS OS FORNECEDORES

### SOLUÇÃO: BASE RELACIONAIS

## MODELO EM REDE

- DESENVOLVIMENTO DO MODELO HIERÁRQUICO PARA DESCREVER AS RELAÇÕES ( **m : n** ) E DIMINUIR O CONSTRANGIMENTO DA HIERARQUIA

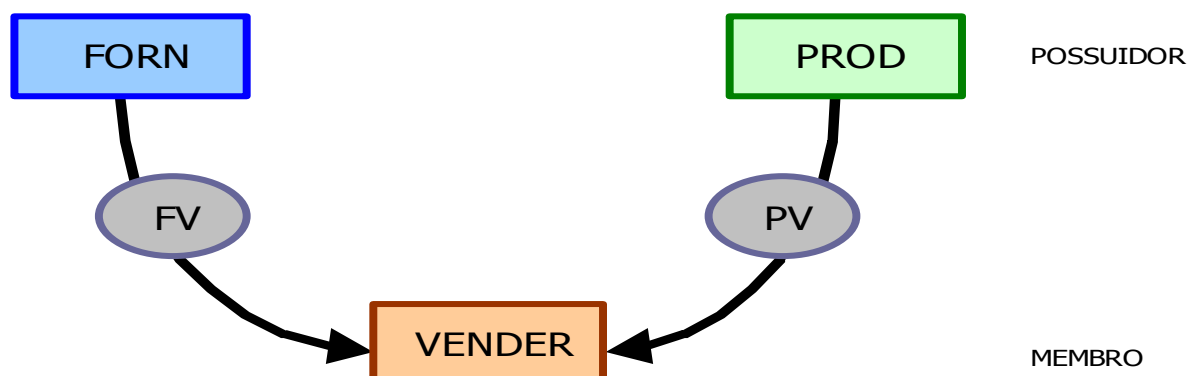
RELAÇÃO: **VENDER**

F_COD	P_COD	V_PUNIT	V_QMIN	V_ATRASO
F1	P1	3	10	15
F1	P2	4	12	30
F1	P3	5	5	0
F2	P1	3	10	15
F2	P3	6	8	15

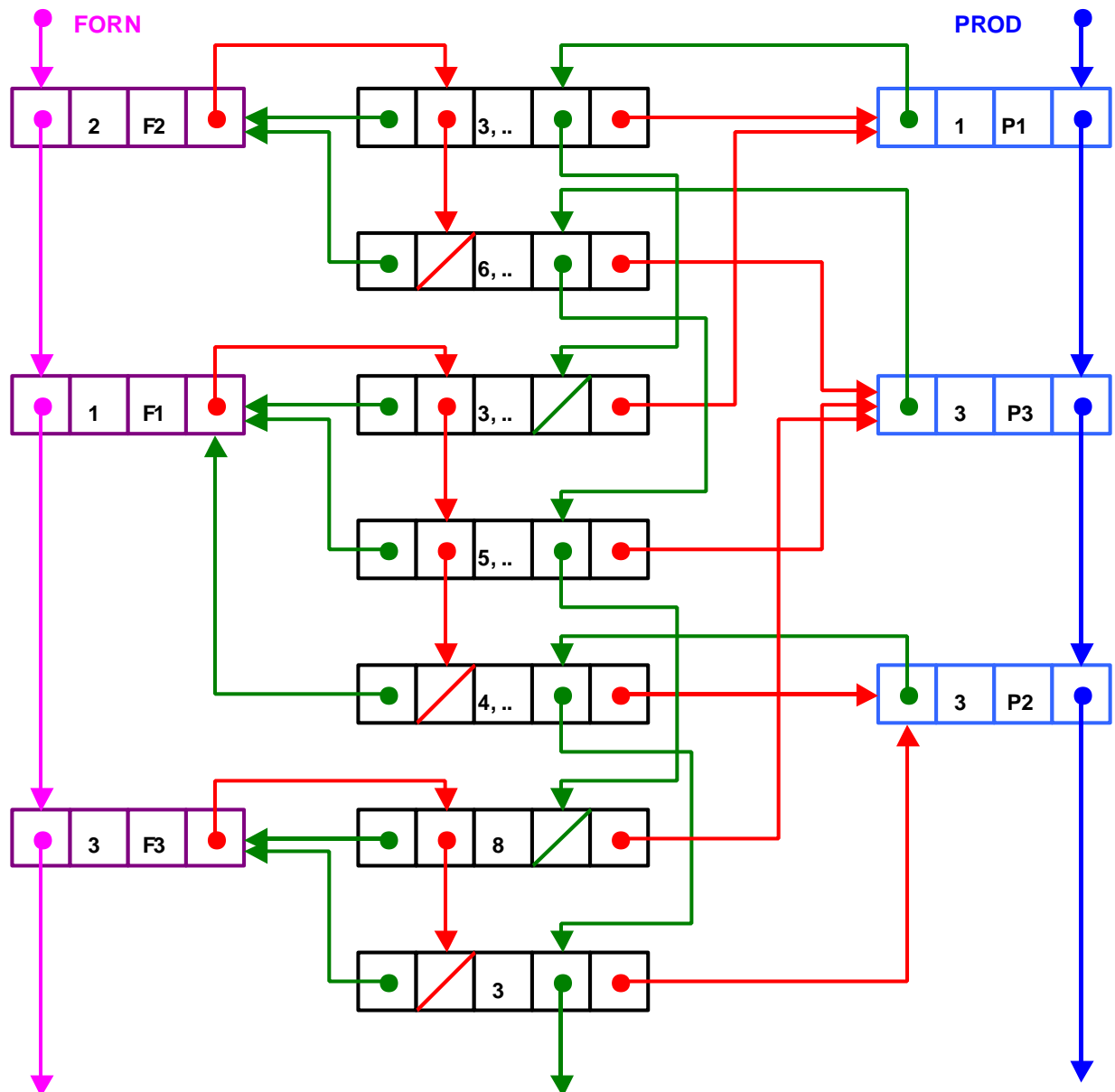
A ESTRUTURA EM REDE PERMITE UTILIZAR RELAÇÕES ( **m : n** )

OS ELEMENTOS DA REDE SÃO OS REGISTOS E NO EXEMPLO ENCONTRAMOS 3 TIPOS: **FORNECEDOR**  
**PRODUTO**  
**VENDER**

PARA DESCREVER A RELAÇÃO ASSOCIAMOS FORNECEDOR A VENDER E VENDER A PRODUTO



## UM REGISTO PODE TER VÁRIOS SUPERIORES HIERÁRQUICOS



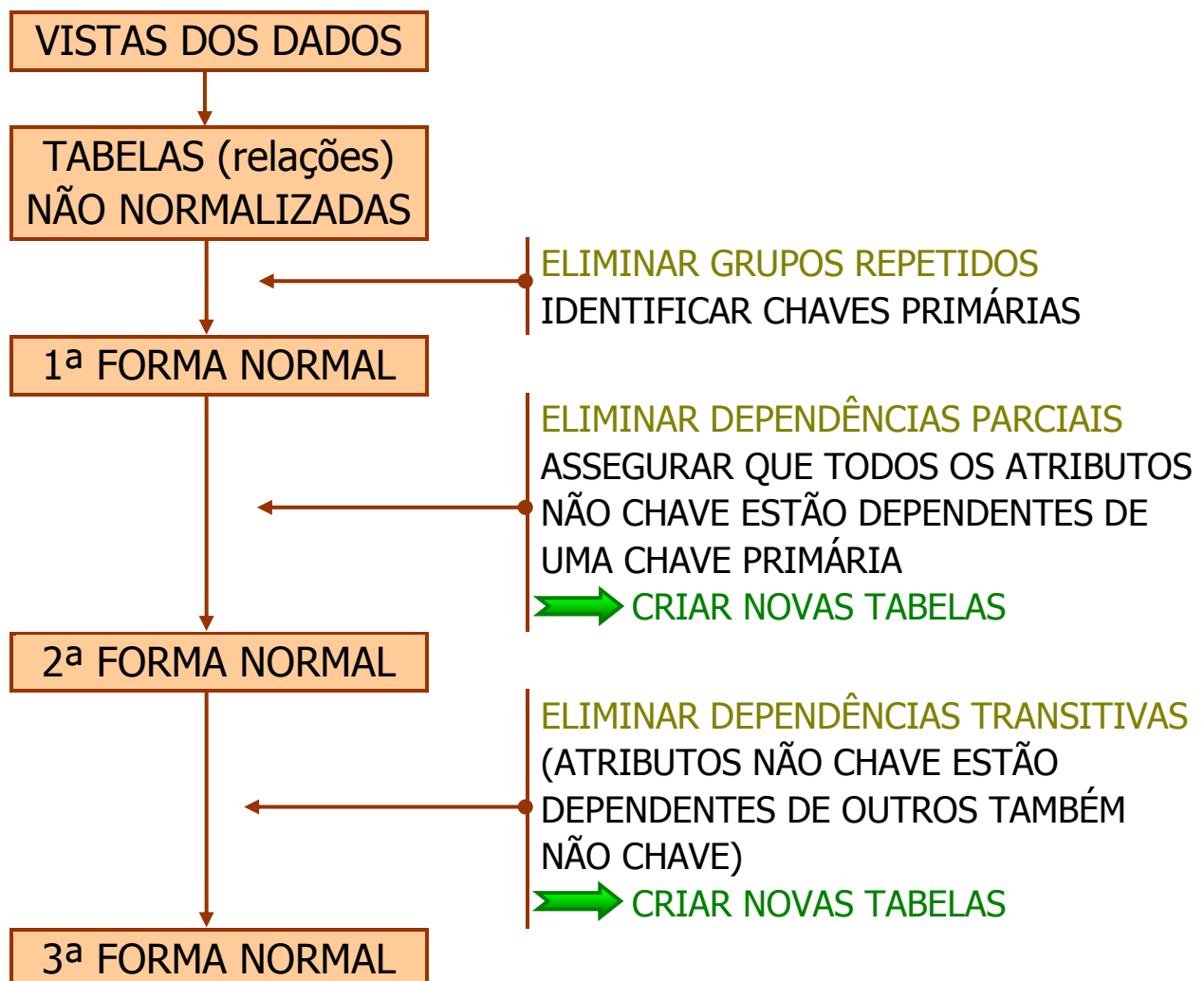
USAM-SE CADEIAS DE APONTADORES

## MODELO RELACIONAL

- ⊕ REPRESENTA A BASE DE DADOS COMO UMA COLECÇÃO DE RELAÇÕES.
- ⊕ CADA RELAÇÃO ASSEMBELHA-SE A UMA **TABELA** DE VALORES.
- ⊕ CADA LINHA DA TABELA REPRESENTA UMA COLECÇÃO DE VALORES RELACIONADOS - **TUPLE**.
- ⊕ O NOME DA COLUMNA DESIGNA-SE POR **ATRIBUTO**.
- ⊕ O TIPO DE DADOS QUE DESCREVE O TIPO DE VALORES EM CADA COLUMNA É DESIGNADO POR **DOMÍNIO**.

PARA QUE O MODELO RELACIONAL SEJA ÚTIL E MANEJÁVEL TEM DE ESTAR NORMALIZADO - NORMALIZAÇÃO:

**TRANSFORMAÇÃO DAS VISÕES COMPLEXAS DOS DADOS E DOS SEUS SUPORTES NUMA COLECÇÃO ESTÁVEL DE ESTRUTURAS SIMPLES**



A alternativa à normalização é: **pensar normalizado!**

## VANTAGENS DE UM SGBD

- ▶ **INDEPENDÊNCIA DOS DADOS**  
Permite um acesso padrão e uma visão arrumada da informação, para que as aplicações possam ser o mais independente possível dos dados
- ▶ **ACESSO A DADOS EFICIENTE E DINÂMICO**  
Inclui estruturas de armazenamento e implementa métodos de acesso e estratégias de otimização genéricas e muito eficientes
- ▶ **INTEGRIDADE E SEGURANÇA DOS DADOS**  
Ajuda na utilização de regras de integridade e controlo de acessos de utilizadores
- ▶ **ADMINISTRAÇÃO DOS DADOS**  
Permite a parametrização de perfis de acesso e estratégias de otimização
- ▶ **ACESSOS CONCORRENTES**  
Prevê a existência de múltiplos utilizadores
- ▶ **RECUPERAÇÃO DE FALHAS**  
Mantêm mecanismos que permitem refazer a informação de forma íntegra
- ▶ **REDUÇÃO TEMPO DESENVOLVIMENTO DE APLICAÇÕES**