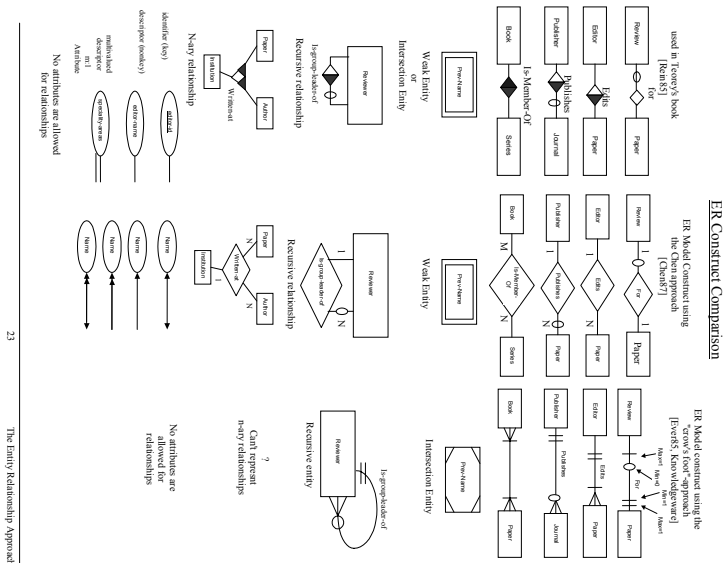


## Comparaç o entre Tipos de Diagramas



## Modelo Relacional

- ▶ Estrutura das Bases de Dados Relacionais
- ▶ Convers o entre Modeliza es
- ▶  lgebra Relacional
- ▶ Opera es Estendidas da  lgebra Relacional
- ▶ Modifica o da Base de Dados
- ▶ Vistas

## Modelos de registos

Os modelos E-R ajudam na modeliza o dos dados. No entanto a maioria dos SGBDs actuais implementa um modelo diferente, embora com um poder descritivo semelhante. Felizmente   f cil passar de um para o outro.

O [Modelo Relacional](#) organiza os dados em:

- ▶ tabelas, ou visto de outra forma, rela es matem ticas.

Este modelo vai permitir responder a:

- ▶ Como   que os dados est o armazenados?
- ▶ Como consultar os dados?
- ▶ Como alterar os dados?

## Exemplo de uma Rela o

numPiso	elevador	portaSaida
6	V	F
-1	F	F
0	V	V
2	V	F
3	V	F
1	V	V
4	V	F
5	V	V

A ordem dos tuplos (linhas da tabela)   irrelevante, o ordenamento   dado simplesmente pela ordem de inser o dos elementos na base de dados.

## Estrutura Básica

- ▶ Formalmente, dados os conjuntos  $D_1, D_2, \dots, D_n$ , uma [relação](#)  $r$  é um subconjunto de  $D_1 \times D_2 \times \dots \times D_n$ .
- ▶ Portanto, uma relação é um conjunto de tuplos  $(a_1, a_2, \dots, a_n)$  com  $a_i \in D_i$ .
- ▶ Os conjuntos  $D_i$  são os [domínios dos atributos](#) pertencentes às entidades e associações que constituem o modelo da base de dados.
- ▶ As relações são então subconjuntos do produto cartesiano dos domínios de variação dos atributos.

Exemplo:

- ▶ domínio de numPiso =  $\mathbb{Z}$ ;
- ▶ domínio de elevador =  $\mathbb{B}$ ;
- ▶ domínio de portaSaida =  $\mathbb{B}$ .

Então:

$$r = \{ (6, V, F), (23, V, V), (-12, F, F) \}$$

é uma relação em numPiso  $\times$  elevador  $\times$  portaSaida

65 / 299

## Atributos

- ▶ Todo o atributo de uma relação tem um nome.
- ▶ O conjunto de valores que um atributo pode tomar é chamado de [domínio do atributo](#).
- ▶ Normalmente, obriga-se a que os valores dos atributos sejam atômicos, ou seja, indivisíveis (1ª Forma Normal):
  - ▶ Por exemplo o atributo endereço é em geral um atributo não atômico (depende das aplicações).
  - ▶ os atributos multi-valor não são atômicos;
  - ▶ os atributos compostos não são atômicos;

66 / 299

## Atributos com Valores Indefinidos

- ▶ Pode-se dar a situação de que numa dada relação nem todos os atributos têm um valor atribuído, por exemplo na relação.

Empregado =

(idEmpregado, nome, telefone, nomesDependentes, inicioFuncoes, idade)

- ▶ Os valores de telefone e nomesDependentes nem sempre estão definidos.
- ▶ Podemos ter tuplos tais como:

(23, João,  $\perp$ ,  $\perp$ , 1970-1-3, 40)

Em termos da terminologia usual em base de dados vai-se designar esse “valor” por null.

- ▶ O valor especial null pertence a todos os domínios.
- ▶ O valor null causa complicações na definição de muitas operações.

Ignoraremos o efeito dos valores nulos em grande parte da apresentação mas consideraremos posteriormente as suas implicações.

67 / 299

## Esquema de Relação

- ▶ Dados os (domínios) atributos  $A_1, A_2, \dots, A_n$ :

$$R = (A_1, A_2, \dots, A_n)$$

é designado por [esquema de relação](#), isto é, é o produto cartesiano de todos os (domínios) atributos.

$$R = \text{EsquemaCliente} = (\text{nome}, \text{endereço}, \text{cidade})$$

- ▶  $r(R)$  é uma [relação](#) (ou [instância de relação](#)) no esquema de relação  $R$ , isto é, um dado subconjunto de  $R$ .

$$\begin{aligned} r(R) &= \text{Cliente}(\text{EsquemaCliente}) \\ &= ('João', 'Rua da Sofia', 'Coimbra') \end{aligned}$$

68 / 299

## Instância de Relação

- ▶ Os valores de uma (instância de) relação são descritos por uma tabela.
- ▶ Um elemento  $t$  de  $r$  é um tuplo, representado por uma linha da tabela.

Por exemplo:

$R = \text{Pisos} = (\text{numPiso}, \text{elevador}, \text{portaSaida})$

$r(R) =$

atributos		
numPiso	elevador	portaSaida
3	V	F
1	V	V
4	V	F
5	V	V

tuplos

69 / 299

## Chaves

O conceito de chave no modelo relacional é em tudo semelhante ao mesmo conceito já visto no modelo entidade-relação.

Seja  $K \subseteq R$ .

- ▶  $K$  é uma **super-chave** de  $R$  se os valores de  $K$  são suficientes para identificar um único tuplo de toda a relação  $r(R)$  possível. Por “relação possível” entende-se uma instância  $r$  que pode existir na empresa que estamos a modelar. Exemplo: {nome, endereço} e {nome}, são ambas super-chaves de Cliente (assumindo-se que não é possível existirem dois clientes com o mesmo nome).
- ▶  $K$  é uma **chave candidata**, se  $K$  é minimal. Exemplo: {nome} é uma chave candidata para Cliente dado ser uma super-chave, e nenhum subconjunto dela é uma super-chave.
- ▶ De entre as chaves candidatas escolhe-se uma delas como sendo a **chave primária** da relação.

70 / 299

## Exemplos de Relações

- ▶ A relação Cliente (Entidade)

nome	endereço	cidade
Pedro	R. Angola 12	Coimbra
João	R. Moçambique	Coimbra

- ▶ A relação ClienteDep (Relação)

nome	número	DataAcesso
Pedro	C-1023	2007-6-11
João	C-304	2007-06-21
Pedro	C-1034	2006-11-1
Isabel	C-304	2005-03-29

71 / 299

## Derivação de relações a partir de um DER

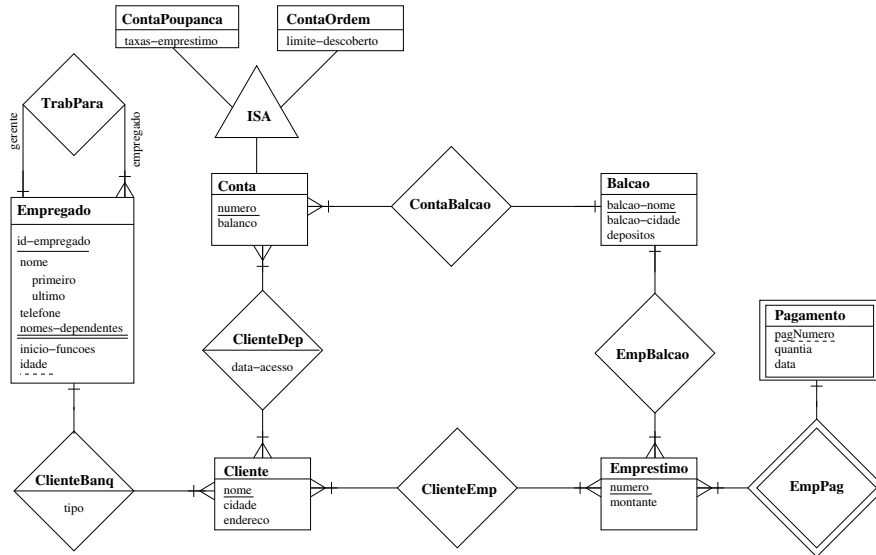
Uma base de dados que seja representável por um DER pode ser também representada por intermédio de um conjunto de relações.

Para cada conjunto de entidades e para cada conjunto de associações gera-se uma única relação (ou tabela) com o nome do conjunto de entidades ou conjunto de associações respectivo.

A conversão de um DER para um esquema de tabelas constitui a base para a derivação do desenho de uma base de dados relacional a partir de um DER.

72 / 299

## DER para um Banco



73 / 299

## Conjuntos de Entidades como Tabelas

Um conjunto forte de entidades reduz-se a uma relação (tabela) com os mesmos atributos.

Balcao
<u>balcaoNome</u>
balcaoCidade
depositos

Balcao(balcaoNome, balcaoCidade, depositos)

74 / 299

## Atributos Compostos

**Atributos Compostos:** cria-se um atributo para cada atributo atômico de um atributo composto.

Por exemplo, considere-se o conjunto de entidades cliente com o atributo composto nome formado por primeiroNome e ultimoNome. A tabela derivada contém os atributos nomePrimeiroNome e nomeUltimoNome.

Cliente
<u>nome</u>
primeiroNome
ultimoNome
cidade
endereco

Cliente(nomePrimeiroNome, nomeUltimoNome, cidade, endereco)

75 / 299

## Atributos Multi-valor

Um **atributo multi-valor** M de uma entidade E é representado através de uma tabela separada EM.

A tabela EM tem os atributos correspondendo à chave primária de E e um atributo correspondendo ao atributo multi-valor M

Por exemplo, o atributo multi-valor nomesDependentes de Empregado é representado pela tabela

EmpregadoNomesDependentes(idEmpregado, nomesDependentes).

Empregado
<u>idEmpregado</u>
nome
telefone
<u>nomesDependentes</u>
inicioFuncoes
idade

Empregado(idEmpregado, nome, telefone, inicioFuncoes, idade)

NomesDependentes(idEmpregado, nomesDependentes)

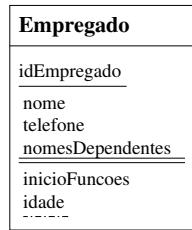
Cada valor de um atributo multi-valor é colocado numa linha separada da tabela EM.

Um empregado com chave primária João e dependentes João e Maria será representado pelas duas linhas: (João, João) e (João, Maria).

76 / 299

## Atributos Derivados

Os **atributos derivados** não têm uma representação directa. Os programas de acesso à informação farão o seu cálculo a partir da informação contida nos atributos dos quais ele é derivado.



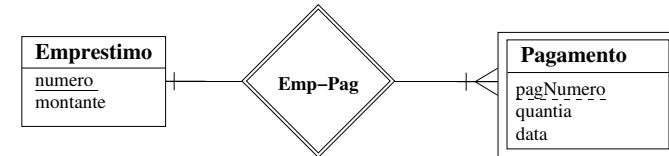
Empregado(idEmpregado, nome, telefone, inicioFuncoes, dataNascimento)

Isto é os atributos derivados são substituídos pelos atributos dos quais eles dependem (idade substituído por dataNascimento) ficando o seu cálculo a cargo dos programas de acesso.

77 / 299

## Conjuntos de Entidades Fracas

Um conjunto de entidades fracas é representado por uma relação que inclui colunas para a chave primária do conjunto de entidades identificador, juntamente com as colunas para os restantes atributos do conjunto de entidades fracas.



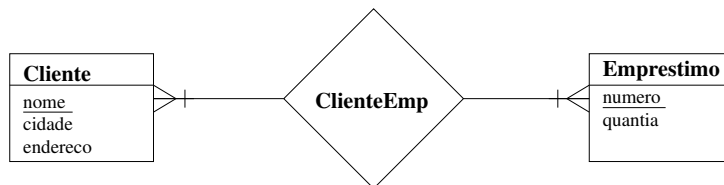
Emprestimo(numero, montante)

Pagamento(pagNumero, quantia, data, numero)

78 / 299

## Conjuntos de Associações

Um conjunto de associações muitos para muitos é representado com uma tabela com colunas para as chaves primárias dos dois conjuntos de entidades participantes, com colunas adicionais para os atributos próprios (ou descritivos) do conjunto de associações.



Cliente(nome, cidade, endereco)

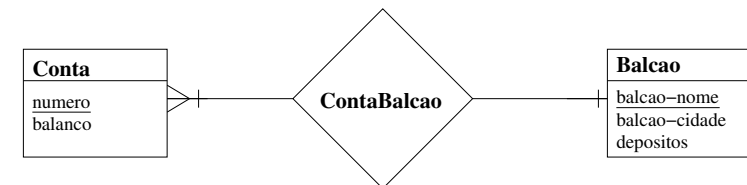
Emprestimo(numero, quantia)

ClienteEmp(nome, numero)

79 / 299

## Tabelas Redundantes

- ▶ Conjuntos de associações muitos-para-um (ou um-para-muitos), totais no lado “muitos” podem ser representados adicionando atributos extra ao lado “muitos”. Entre esses atributos tem de estar contida a chave primária do outro conjunto participante.
- ▶ Por exemplo: Em vez de se criar uma tabela para a associação ContaBalcao, adiciona-se a coluna balcaoNome à tabela derivada a partir do conjunto de entidades Conta.



Conta(conta, balanco, balcaoNome)

Balcao(balcaoNome, balcaoCidade, depositos)

80 / 299

## Redundância de Tabelas (Cont.)

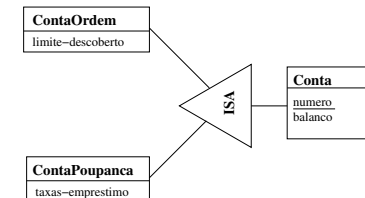
- ▶ Para conjuntos de associações um-para-um, qualquer dos lados pode receber a chave primária do outro lado.
- ▶ Se a associação é parcial no lado muitos, a substituição da tabela por uma coluna extra pode levar à ocorrência de valores nulos.
- ▶ É redundante a tabela correspondente ao conjunto de associações relacionando um conjunto de entidades fracas com o seu conjunto identificador.  
Por exemplo: A tabela Pagamento já contém a informação que apareceria na tabela EmpPag (as colunas numeroEmprestimo e numeroPagamento).

81 / 299

## Derivação de Tabelas para a Especialização

### Método 1: “Descendente”

- ▶ Formar uma tabela para a entidade de maior nível (mais geral).
- ▶ Criar uma tabela para cada conjunto de entidades de nível abaixo, incluindo a chave primária da entidade acima e os atributos locais.



Conta(numero, balanço)  
ContaOrdem(numero, limiteDescoberto)  
ContaPoupanca(numero, taxasEmprestimo)

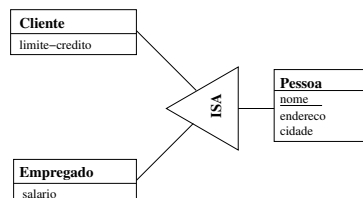
**Desvantagem:** obter a informação acerca de ContaPoupanca (por exemplo) obriga à consulta de duas tabelas

82 / 299

## Derivação de Tabelas para a Especialização

### Método 2: “Ascendente”.

- ▶ Formar uma tabela para cada conjunto de entidades com os atributos locais e herdados.
- ▶ Se a especialização é total, não há necessidade de criar uma tabela para a entidade mais geral (Pessoa)



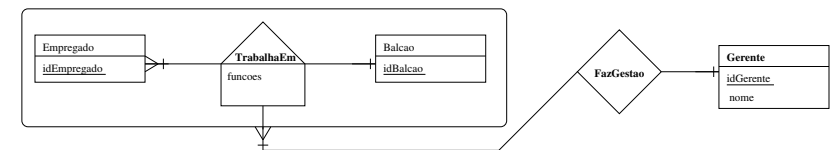
Cliente(nome, endereco, cidade, limiteCredito)  
Empregado(nome, endereco, cidade, salario)

**Desvantagem:** no caso de especializações não disjuntas pode ocorrer duplicação de informação. Por exemplo, no caso de um empregado que também seja um cliente, o endereço e o nome da cidade são duplicados.

83 / 299

## Relações Correspondendo à Agregação

Para representar agregações,



criar uma tabela com:

- ▶ a chave primária da associação agregada (TrabalhaEm);
- ▶ a chave primária do conjunto de entidades participante (Gerente);
- ▶ Restantes atributos descritivos (FazGestao).

FazGestao(idGerente, idEmpregado, idBalcao)

Dentro da agregação aplicam-se as regras já descritas acima.

84 / 299

## Determinação de Chaves a partir do DER

- ▶ Conjunto de entidades fortes. A chave primária do conjunto de entidades é a chave primária da relação.
- ▶ Conjunto de entidades fracas. A chave primária da relação consiste na união da chave primária do conjunto de entidades forte com o discriminante do conjunto de entidades fracas.
- ▶ Conjunto de relações. A união das chave primárias dos conjuntos de entidades relacionados é uma super-chave da relação.
  - ▶ Para conjuntos de associações binários um-para-muitos, a chave primária do lado “muitos” é a chave primária da relação.
  - ▶ Para conjuntos de associações um-para-um, a chave primária da relação é a chave primária de um dos conjuntos de entidades.
  - ▶ Para conjuntos de associações muitos-para-muitos, a união das chaves primárias é a chave primária da relação.

85 / 299

## Modelo Relacional para o Banco (simplificado)

ContaPoupanca(numero, taxaEmprestimo)

ContaOrdem(numero, limiteDescoberto)

Conta(numero, balanco, balcaoNome)

Balcao(balcaoNome, balcaoCidade, depositos)

ClienteDep(nome, numero, dataAcesso)

Cliente(nome, cidade, endereco)

ClienteEmp(nome, numero)

Emprestimo(numero, quantia, balcao)

86 / 299

## Linguagem de Consulta/Interrogação

Linguagem a que o utilizador recorre para obter informação a partir da base de dados.

Categorias de linguagens

- ▶ Procedimentais
- ▶ Declarativas

Linguagens Teóricas

- ▶ Álgebra Relacional
- ▶ Cálculo Relacional de Tuplos
- ▶ Cálculo Relacional de Domínios

Estas linguagens formam a base teórica das linguagens de consulta utilizadas na prática.

87 / 299

## Álgebra Relacional

- ▶ Linguagem declarativa
- ▶ Seis operadores básicos
  - ▶ selecção — definir condições para as quais se quer obter a informação contida numa dada relação.
  - ▶ projecção — seleccionar, numa relação, quais os atributos que se quer visualizar.
  - ▶ união
  - ▶ diferença de conjuntos
  - ▶ produto cartesiano
  - ▶ renomeação
- ▶ Os operadores têm como argumentos relações de entrada e devolvem uma relação como resultado.

88 / 299