



Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA  
Departamento de Informática  
Integrado / Análise e Desenvolvimento de Sistemas / Licenciatura em  
Computação

## Modelo Lógico

André L. R. Madureira <[andre.madureira@ifba.edu.br](mailto:andre.madureira@ifba.edu.br)>  
Doutorando em Ciência da Computação (UFBA)  
Mestre em Ciência da Computação (UFBA)  
Engenheiro da Computação (UFBA)

# Modelo Lógico

---

- É o mais próximo de uma implementação de projeto físico de banco de dados
- Existem vários modelos, porém o mais usado é o **modelo relacional**, baseado em tabelas

Tabela de Clientes			
ID	<u>Nome</u>	<u>CPF</u>	<u>ID Conta</u>
1	Julia	111.222.333-44	2
2	Carlos	555.666.777-88	1
3	Amanda	123.456.789-00	3

Tabela de Contas		
ID	<u>Agência</u>	<u>Número</u>
1	3460	71542
2	5421	65321
3	7410	02145

# Modelo Relacional

---

- No modelo relacional, dados são armazenados no formato de tabelas (**relações**)
- Cada atributo é uma coluna e cada instância é uma linha
- Não podemos ter instâncias exatamente iguais (todos os atributos idênticos)

Tabela de Clientes			
ID	<u>Nome</u>	<u>CPF</u>	<u>ID Conta</u>
1	Julia	111.222.333-44	2
2	Carlos	555.666.777-88	1
3	Amanda	123.456.789-00	3

# Representação com Diagrama de Esquemas

---

- Representar todos os atributos e instâncias em uma tabela ocupa muito espaço
  - **Solução:** descrever apenas o esquema da tabela usando o **diagrama de esquemas**
    - **Ex:**

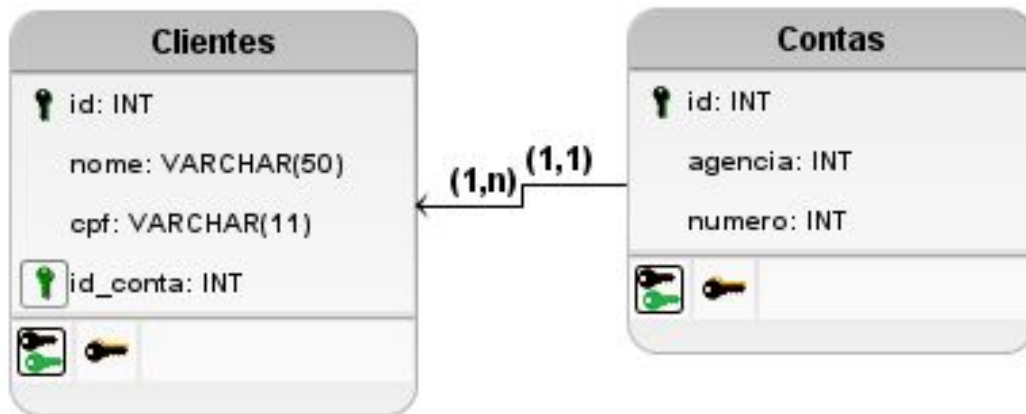
## DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

**Esquema** é a estrutura da tabela (ex: nome, atributos, chaves, relacionamentos, etc)

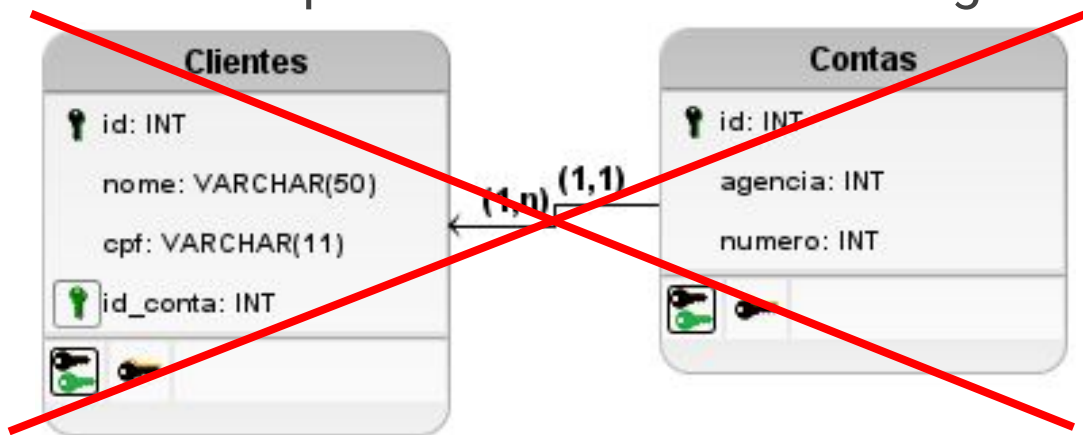
# Representação “UML”

- Outra forma de representar o esquema de tabelas é usando a notação da ferramenta **brModelo**
  - Notação semelhante ao diagrama de classes UML



# Representação matemática

- **Problema:** Descrever o esquema da tabela de forma gráfica ocupa muito espaço



- **Solução:** descrever o esquema usando notação matemática
  - **Ex:**  $\text{Clientes}(\text{id}, \text{nome}, \text{cpf}, \text{id\_conta})$   
 $\text{Contas}(\text{id}, \text{agencia}, \text{numero})$

# Representação matemática

- **Sintaxe:**  $R(A_1, A_2, \dots, A_n)$ 
  - **R** é o nome da relação
  - **$A_i$**  é o nome do atributo
  - **n** é o grau ou aridade da relação
  - **$\text{dom}(A_i)$**  é o domínio do atributo  **$A_i$**
- **Ex:** Contas(id, agencia, numero)
  - A relação Contas tem atributos id, agencia e numero
  - A relação Contas tem grau 3



# Representação matemática

---

- **Ex:** Clientes(id, nome, cpf, id\_conta)
  - A relação Clientes tem atributos id, nome, cpf e id\_conta
  - A relação Clientes tem aridade 4
  - $\text{dom}(\text{id}) = \text{INT}$
  - $\text{dom}(\text{nome}) = \text{VARCHAR}(50)$
  - $\text{dom}(\text{cpf}) = \text{VARCHAR}(11)$
  - $\text{dom}(\text{id\_conta}) = \text{INT}$





# Representação matemática

- Uma relação **R** possui estados **r(R)** que representam que dados estão armazenados em um determinado momento nessa relação
- **r(R) = {t<sub>1</sub>, t<sub>2</sub>, ..., t<sub>m</sub>}** onde **t<sub>k</sub> = <v<sub>1</sub>, v<sub>2</sub>, ..., v<sub>n</sub>>** é uma tupla
  - **v<sub>i</sub>** são os valores de cada atributo **A<sub>i</sub>** da tupla **t**
  - **v<sub>i</sub> ∈ dom(A<sub>i</sub>)**
- **Ex:** Seja R = Clientes(id, nome, cpf, id\_conta)
  - t<sub>1</sub> = <1, 'Julia', 111.222.333-44, 2>
  - t<sub>2</sub> = <2, 'Carlos', 555.666.777-88, 1>
  - t<sub>3</sub> = <3, 'Amanda', 123.456.789-00, 3>

Clientes			
ID	Nome	CPF	ID Conta
1	Julia	111.222.333-44	2
2	Carlos	555.666.777-88	1
3	Amanda	123.456.789-00	3

# Representação matemática

---

- Podemos acessar o valor de cada atributo de uma tupla usando a notação  $t[A_i]$  ou  $t.A_i$
- **Ex:** Seja  $R = \text{Clientes}(\text{id}, \text{nome}, \text{cpf}, \text{id\_conta})$  e  $r(R) = \{t_1, t_2\}$ 
  - $t_1 = \langle 1, \text{'Julia'}, 111.222.333-44, 2 \rangle$
  - $t_2 = \langle 2, \text{'Carlos'}, 555.666.777-88, 1 \rangle$
- Podemos acessar os valores dos atributos de  $t_i$ :
  - $t_1[\text{nome}] = \langle \text{'Julia'} \rangle$
  - $t_1.\text{nome} = \langle \text{'Julia'} \rangle$
  - $t_2[\text{cpf}, \text{id\_conta}] = \langle 555.666.777-88, 1 \rangle$

# Exercício - Representação matemática

---

- Seja um sistema de livraria online, com as seguintes relações:
  - $R_1 = \text{Clientes}(\text{id}, \text{nome}, \text{cpf}, \text{endereco})$
  - $R_2 = \text{Livros}(\text{id}, \text{titulo}, \text{autor}, \text{editora}, \text{genero})$
- Forneça exemplos de tuplas de cada uma das relações acima
- Explique porque não é possível inserir duas tuplas idênticas. Forneça exemplos.

# Tipo de Dados de Atributos

---

- Define que valores de dados um atributo pode assumir (**domínio do atributo**)
- Cada sistema de gerenciamento de banco de dados (SGBD) fornece suporte a tipos de dados diferentes
  - int, smallint, bigint
  - float, double
  - boolean
  - date, datetime, time
  - varchar(n), text

# INT ou INTEGER

---

- Inteiro de tamanho normal (4 bytes),
  - Um inteiro **com** sinal pertence ao intervalo:
    - $[-2^{31}, +2^{31}-1] = [-2147483648, +2147483647]$
  - Um inteiro **sem** sinal pertence ao intervalo:
    - $[0, 2^{32}] = [0, 4294967295]$
- Podemos ter números inteiros de tamanho maior ou menor que 4 bytes, de acordo com o que for necessário para a aplicação
  - TINYINT (1B), SMALLINT (2B), INT (4B), BIGINT (8B)

# INT ou INTEGER

---

- Cálculo do intervalo de valores de um inteiro de tamanho N bytes:
  - Se o inteiro possui sinal (+/-):
    - $[-2^{N*8}/2, +2^{N*8}/2 - 1]$
  - Se o inteiro **não** possui sinal (+/-):
    - $[0, +2^{N*8}]$

# FLOAT, DOUBLE e NUMERIC(p,d)

---

- Computadores não conseguem armazenar um conjunto de valores infinito com exatidão, pois eles são sistemas binários
  - Porém números fracionários são infinitos
  - **Problema:** ao fazer operações com números fracionários, podemos ter erros de aproximação
  - Erros de aproximação são minimizados pela precisão do número fracionário (FLOAT ou DOUBLE)

# FLOAT, DOUBLE e NUMERIC(p,d)

---

- **Float:** Número decimal de precisão simples (32 bits)
- **Double:** Número de decimal de precisão dupla (64 bits)
- **Numeric(p,d):** número decimal de ponto fixo com **p** dígitos (incluindo a parte inteira e fracionária) e **d** casas decimais (algarismos da parte fracionária)
  - **Ex:** numeric(3,1) permite armazenar os números:
    - 44,2
    - 1,8



# BOOLEAN

---

- **BOOLEAN:** Valores booleanos que podem ser representado por false/true ou 0/1
  - **Ex:** Aprovação ou reprovação em uma disciplina

```
create table disciplina (  
    nome_aluno varchar(20),  
    nota double,  
    aprovado_reprovado boolean  
);
```

# DATE e TIME

---

- **DATE:** Uma data. A faixa suportada é entre '1000-01-01' e '9999-12-31'
  - MySQL mostra valores DATE no formato 'AAAA-MM-DD'
  - **Ex:** armazenar data de nascimento de uma pessoa
- **TIME:** A faixa é entre '-838:59:59' e '838:59:59'.
  - MySQL mostra valores TIME no formato 'HH:MM:SS'
  - **Ex:** armazenar o horário de chegada e saída no IFBA

# DATETIME

---

- **DATETIME:** Um combinação de hora e data. A faixa suportada é entre '1000-01-01 00:00:00' e '9999-12-31 23:59:59'
  - MySQL mostra valores DATETIME no formato 'AAAA-MM-DD HH:MM:SS'
  - **Ex:** armazenar o dia e horário que um funcionário chegou para trabalhar

# VARCHAR e TEXT

---

- **VARCHAR (M):** Uma string de tamanho variável **M** caracteres
  - $M = [1, 255]$
  - Se o valor M especificado for maior que 255, o tipo do atributo é convertido para **TEXT**
- **TEXT:** String com tamanho máximo de 65535 caracteres
  - Cada valor do atributo consome  $2^{16}-1$  bytes
- **LONGTEXT:** máximo de 4,294,967,295 caracteres
  - Cada valor do atributo consome  $2^{32}-1$  bytes

# Chave em Banco de Dados

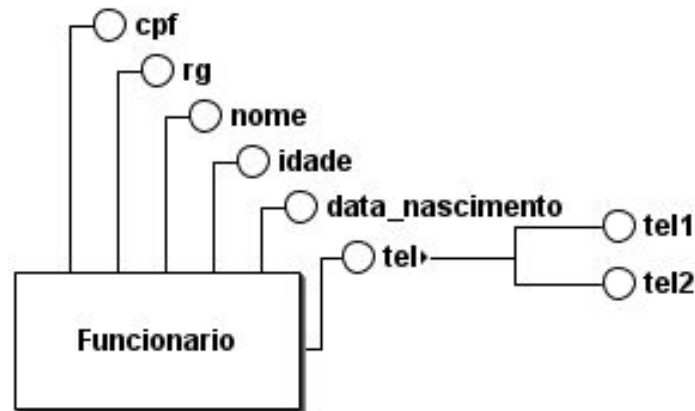
---

- É um ou mais atributos de uma tabela que identificam um registro ou conjunto de registros de forma exclusiva
  - **Chave candidata**
    - **Chave primária**
      - **Chave primária surrogada**
  - **Chave estrangeira**

# Chave Candidata

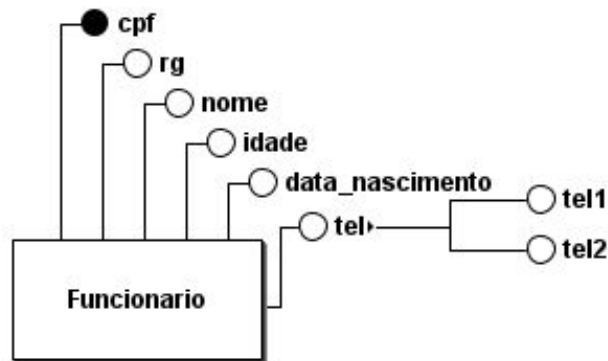
- São **atributos determinantes** que PODEM identificar registros de uma tabela de forma exclusiva
  - Não pode ter valores repetidos ou nulos  
**(restrição de integridade de entidade)**

- Funcionários diferentes possuem CPFs e RGs distintos
- Podemos usar CPF e/ou RG para identificar cada funcionário



# Chave Primária (*Primary Key* ou *PK*)

- É uma **chave candidata** ESCOLHIDA pelo projetista do banco de dados para identificar os registros



**Representação no modelo E-R:**  
círculo preenchido



**Representação no modelo lógico:**  
atributo sublinhado, ou com chave dourada ou preta ao lado

# Chave Primária Surrogada

- Não possui significado para aplicação ou usuário
- Serve apenas para garantir a exclusividade dos registros
- Normalmente esse atributo é escondido do usuário

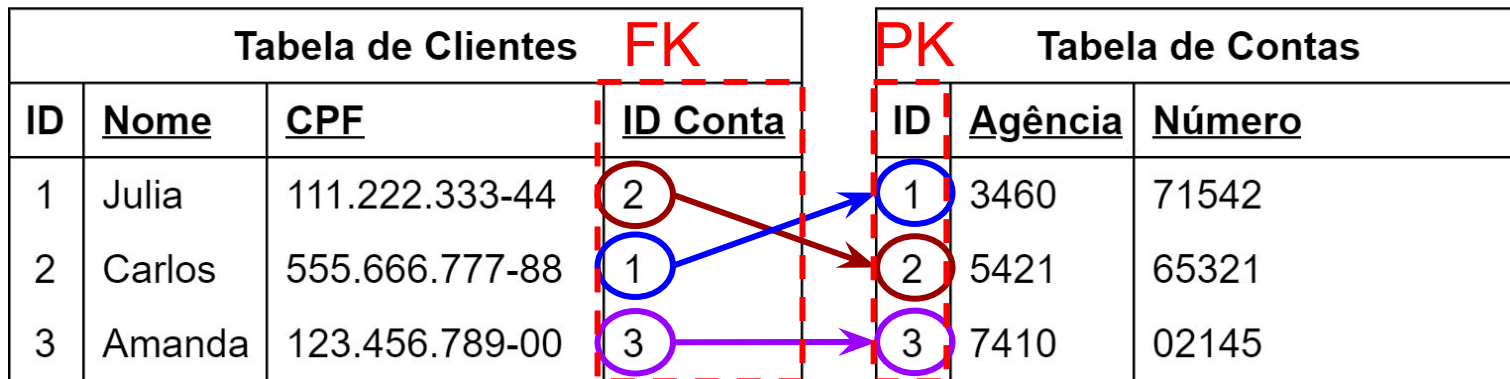
Tabela de Clientes			
<u>ID</u>	<u>Nome</u>	<u>CPF</u>	<u>ID Conta</u>
1	Julia	111.222.333-44	2
2	Carlos	555.666.777-88	1
3	Amanda	123.456.789-00	3

Tabela de Contas		
<u>ID</u>	<u>Agência</u>	<u>Número</u>
1	3460	71542
2	5421	65321
3	7410	02145

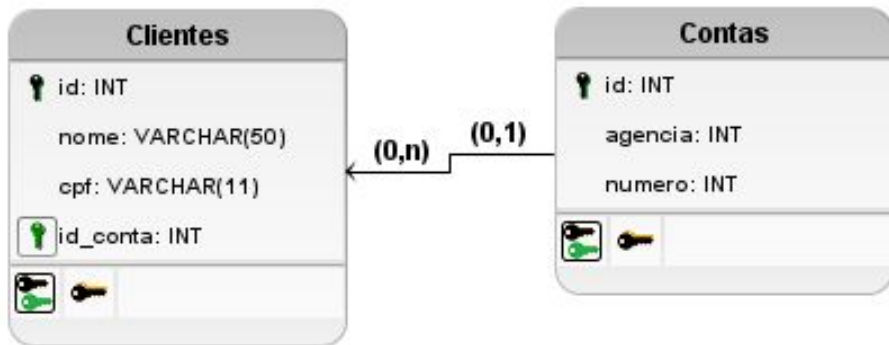


# Chave Estrangeira (*Foreign Key* ou *FK*)

- É um atributo de uma tabela que estabelece um relacionamento com a chave primária de outra tabela
- Permite buscar dados armazenados em múltiplas tabelas



# Chave Estrangeira (*Foreign Key* ou *FK*)



**Representação no modelo lógico:**  
atributo com chave prata ou verde ao lado

Dizemos que o atributo “*Cliente.id\_conta*” é uma chave estrangeira que **referencia** o atributo “*Contas.id*”

Outra forma de falar é que “*Cliente.id\_conta*” **refere-se** ao atributo “*Contas.id*”

Isto é, há **restrição de integridade referencial** de Clientes para Contas

# Chave Estrangeira (*Foreign Key* ou *FK*)

Uma chave estrangeira pode se referir a um atributo de sua mesma tabela

**Ex:** cpf\_supervisor em Funcionário se refere a um cpf da mesma tabela Funcionário

## FUNCIONARIO

Pnome	Minicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Alice	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 85, São Paulo, SP	M	55.000	NULL	1

**Chaves estrangeiras podem conter o valor NULL**

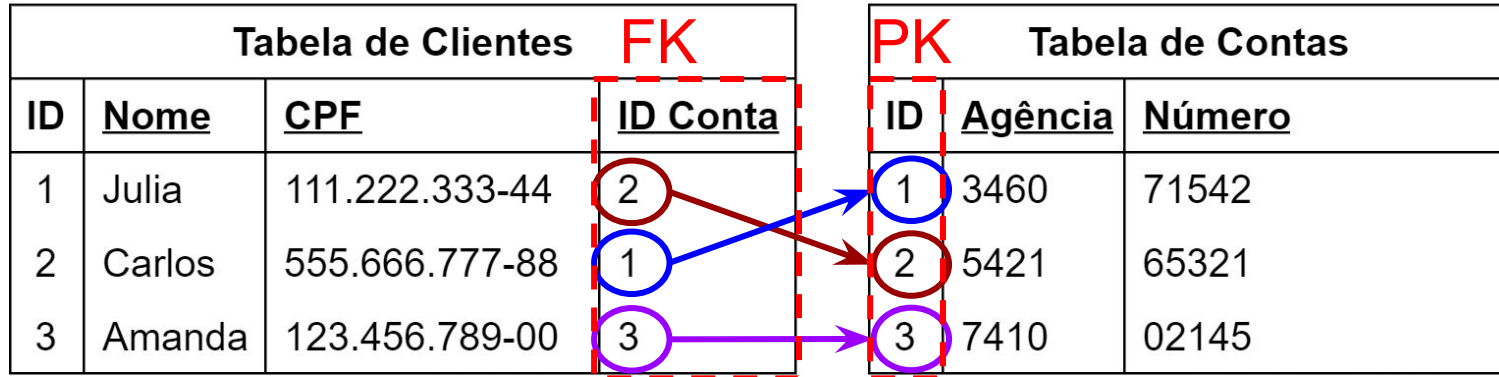
# Exercício - Chaves

---

- Seja um sistema de vendas online, onde temos:
  - Clientes que compram um ou mais produtos
  - Produtos precisam ser entregues no endereço do cliente
  - Ha um frete a ser pago e um prazo de entrega
  - Funcionários organizam as entregas para cada cidade
- Quais são as chaves primárias e estrangeiras das tabelas desse sistema?
- Porque não podemos ter valores NULL nas chaves primárias?

# Relacionamentos no Modelo Lógico

- Chaves estrangeiras e primárias são usadas para estabelecer relacionamentos
  - Chave estrangeira => chave primária

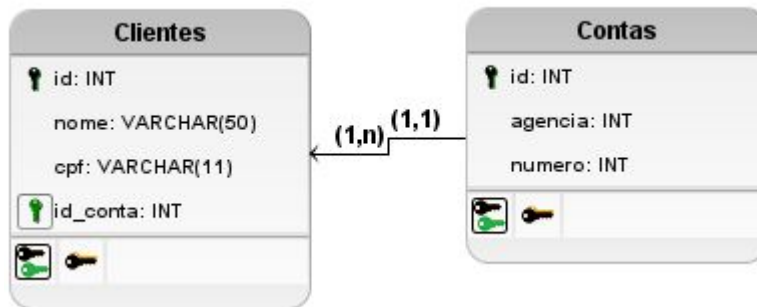


# Relacionamentos 1:N (um para muitos)

Podemos ter **N** clientes associados a **1** mesma conta

Clientes				Contas		
id	nome	cpf	id_conta	id	agencia	numero
1	João	111.222.333-44	4	4	1234	1111
2	Josefa	444.555.666-77	5	5	5678	2222
3	Carlos	888.999.000-11	4	6	9102	3333
4	Maria	123.456.789-12	6	7	3456	4444

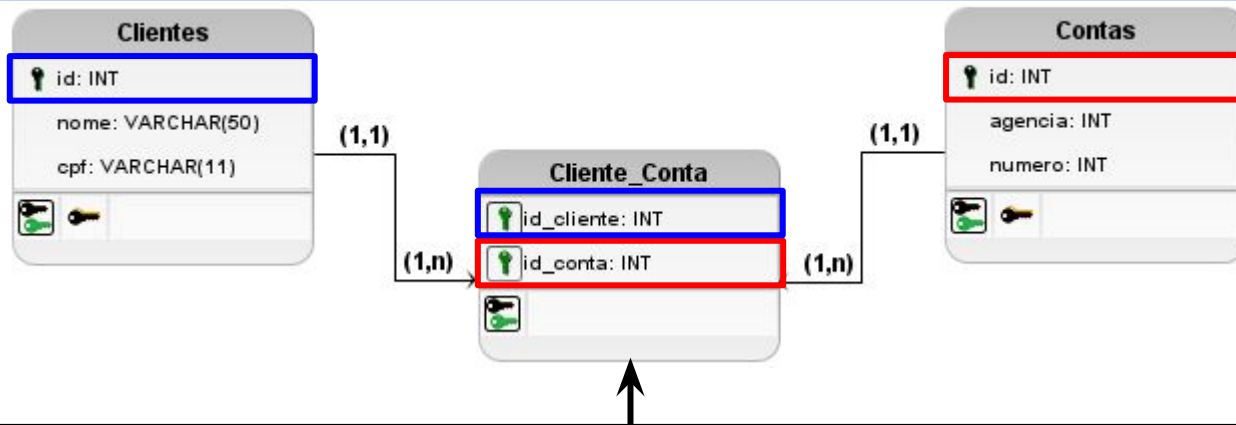
**Representação no modelo lógico:**  
seta com uma das extremidades com cardinalidade máxima N



Cada cliente pode ter mais de uma conta no banco.

Como representar isso no DB?

# Relacionamentos N:N (muitos para muitos)



## **Tabela de referência cruzada (ou de Junção):**

Tabela associativa que mapeia duas ou mais tabelas através de suas chaves primárias

A tabela é composta pelas chaves estrangeiras das tabelas referenciadas

**Convenção:** nome da tabela formado pelo nome das tabelas referenciadas

# Relacionamentos N:N (muitos para muitos)

Cada 1 cliente pode possuir N contas no banco

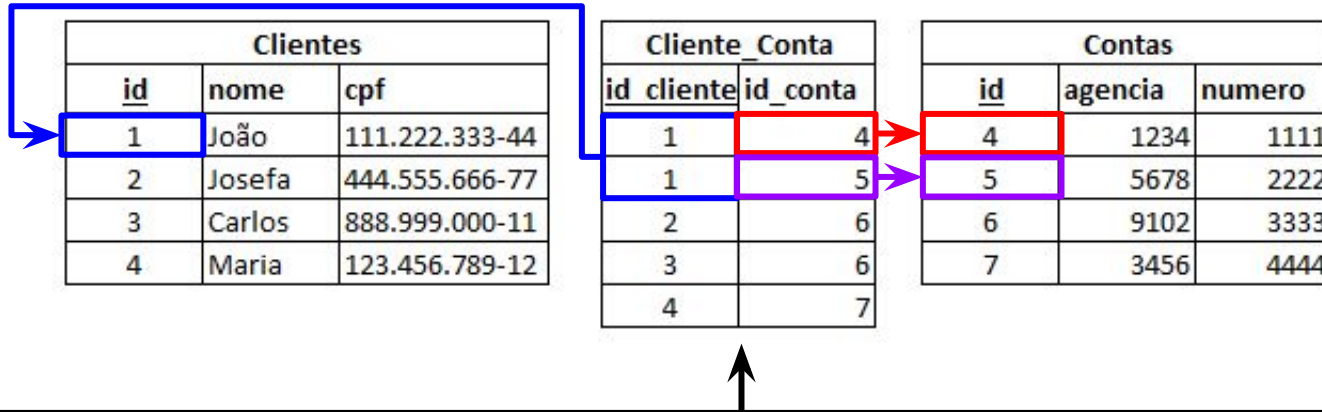


Tabela de referência cruzada (ou de Junção)



# Relacionamentos N:N (muitos para muitos)

E **N** clientes podem possuir **a mesma** conta no banco

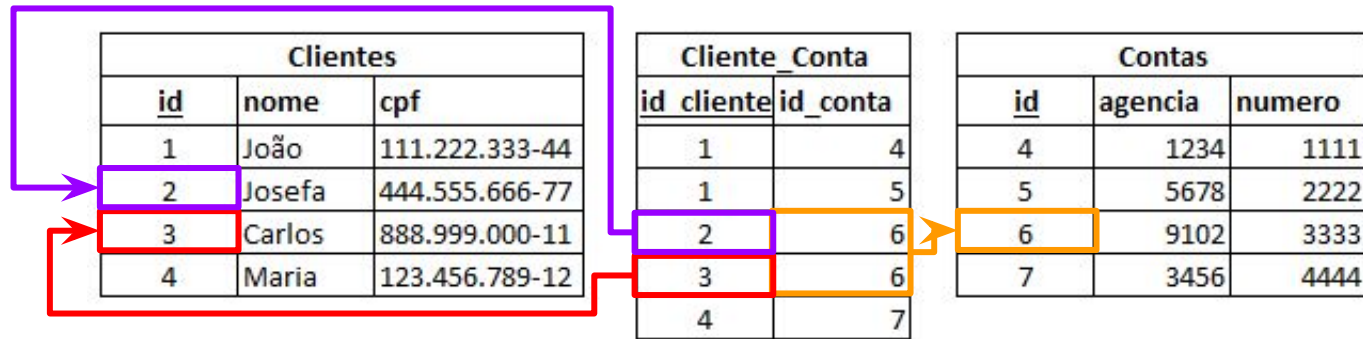


Tabela de referência cruzada (ou de Junção)

# Exercício - Relacionamentos

---

- Construa um modelo lógico para um sistema bancário, no qual:
  - Cada gerente pode gerenciar mais de uma conta
  - Existem diferentes tipos de conta (corrente ou poupança)
  - Cada cliente pode ter mais de uma conta no banco
- Forneça exemplos de como as tabelas se relacionam entre si

# Como traduzir do Modelo Conceitual para o Modelo Relacional?

---

- Tivemos um trabalho considerável para criar o modelo conceitual para facilitar a criação do modelo relacional do DB, mas como conseguimos traduzir de um modelo para o outro ?
  - Devemos seguir a sequência de passos:
    - Tradução das entidades em tabelas (relações)
    - Tradução dos relacionamentos 1:1 e 1:N binários
    - Tradução dos relacionamentos N:N e relacionamentos n-ários
    - Tradução dos atributos multivalorados ou repetitivos
    - Tradução dos entidades especializadas

# Tradução de Entidades e Relacionamentos 1:1 ou 1:N

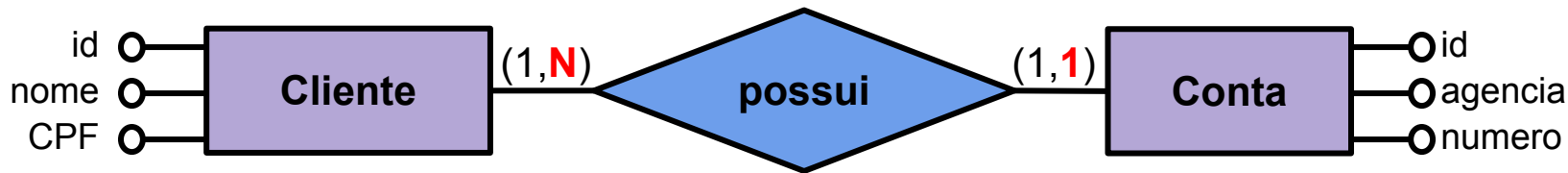
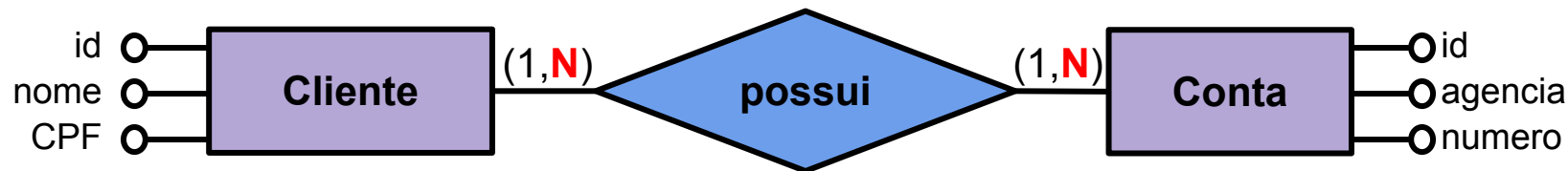


Tabela de Clientes				Tabela de Contas		
ID	Nome	CPF	ID Conta	ID	Agência	Número
1	Julia	111.222.333-44	2	1	3460	71542
2	Carlos	555.666.777-88	1	2	5421	65321
3	Amanda	123.456.789-00	3	3	7410	02145

# Tradução de Relacionamentos N:N



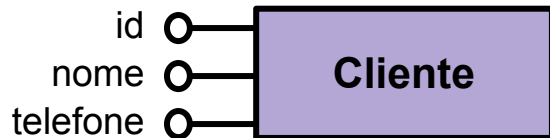
Clientes		
id	nome	cpf
1	João	111.222.333-44
2	Josefa	444.555.666-77
3	Carlos	888.999.000-11
4	Maria	123.456.789-12

Cliente_Conta	
id_cliente	id_conta
1	4
1	5
2	6
3	6
4	7

Contas		
id	agencia	numero
4	1234	1111
5	5678	2222
6	9102	3333
7	3456	4444

Tabela de junção

# Tradução de Atributos Multivalorados



Cliente		
id	nome	telefone
1	João	99115-4584
2	Josefa	3214-5869
3	Carlos	NULL
4	João	98561-5869

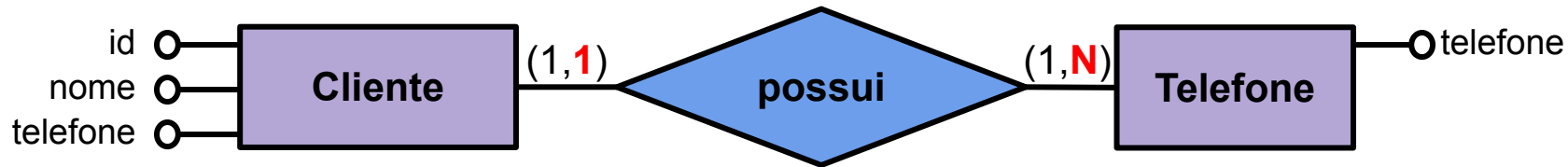
## Problema 1:

Como telefone é multivalorado, precisamos **repetir todos os outros atributos** somente para adicionar um novo telefone.

## Problema 2:

Pessoas sem telefone tem atributo NULL.

# Tradução de Atributos Multivalorados



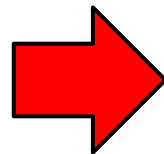
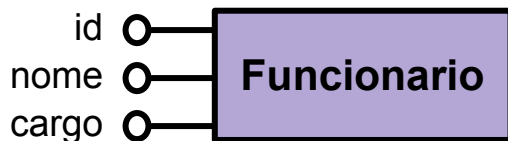
Para cada atributo multivalorado,  
criamos uma tabela nova **S**

Relacionamos **S** com a tabela  
original (detentora do atributo)  
usando uma chave estrangeira

Cliente	
<u>id</u>	nome
1	João
2	Josefa
3	Carlos

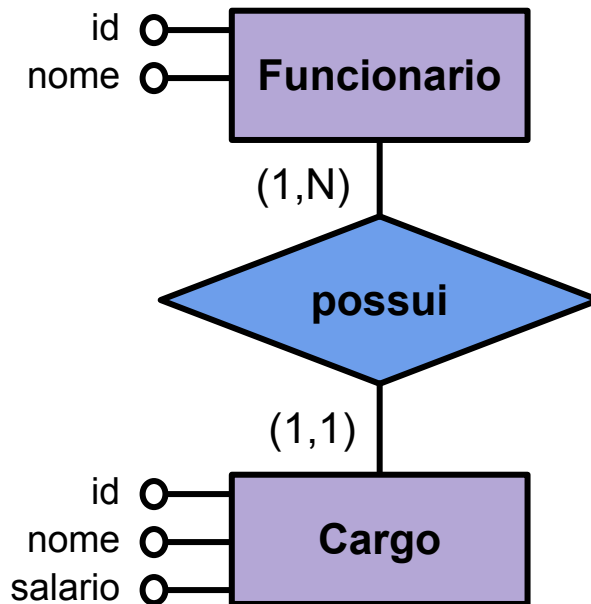
Telefone_Cliente	
<u>id_cliente</u>	<u>telefone</u>
1	99115-4584
1	98561-6541
2	3214-5869

# Tradução de Atributos Repetitivos



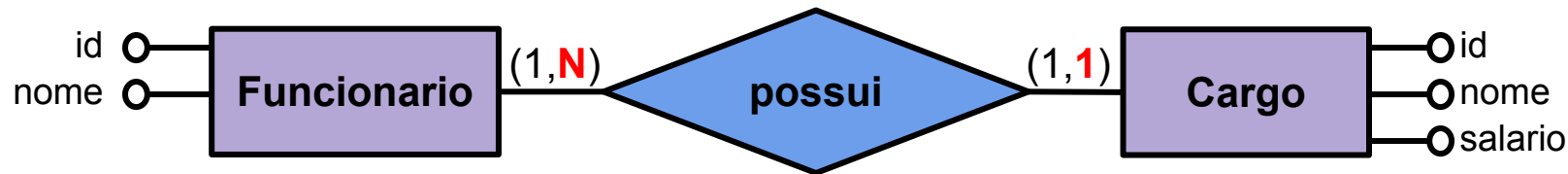
Para cada atributo repetitivo, criamos uma nova entidade e um relacionamento binário

Alguns autores chamam essa nova entidade de **entidade (ou tabela) de cadastro**





# Tradução de Atributos Repetitivos

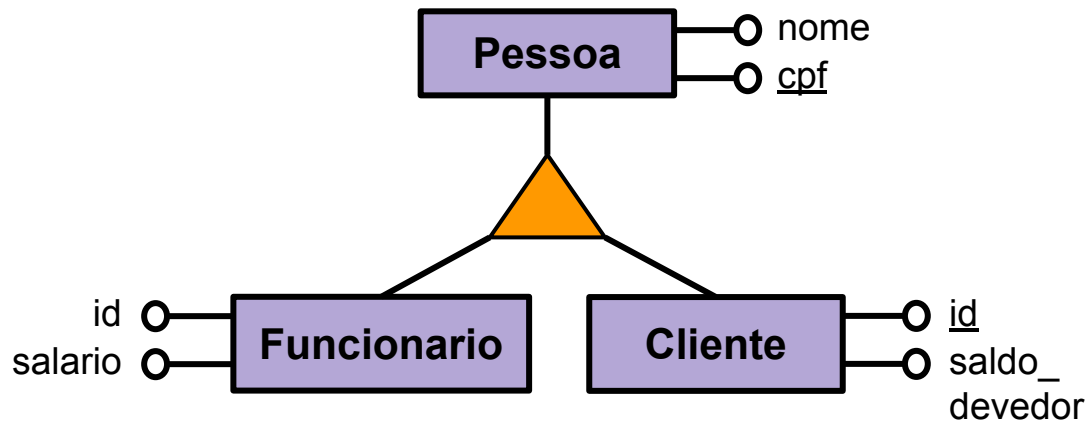


Agora podemos traduzir para o modelo relacional usando a **regra dos relacionamentos 1:N**

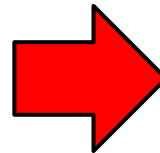
Funcionario			Cargo_Cliente	
<u>id</u>	nome	id_cargo	<u>id</u>	<u>cargo</u>
1	João	1	1	Padeiro
2	Josefa	1	2	Caixa
3	Carlos	2		Gerente

Diagram illustrating the translation of the 1:N relationship into a relational model. The **Funcionario** table has attributes **id** (primary key), **nome**, and **id\_cargo**. The **Cargo\_Cliente** table has attributes **id** (primary key) and **cargo**. Red boxes highlight the mapping of **id\_cargo** values from **Funcionario** to **id** in **Cargo\_Cliente**. A blue box highlights the **id** value 2 in **Cargo\_Cliente**, which corresponds to the **id\_cargo** value 2 in **Funcionario**. A blue arrow points from the **id\_cargo** value 2 in **Funcionario** to the **id** value 2 in **Cargo\_Cliente**.

# Tradução de Entidades Especializadas



Para cada entidade especializada, criamos uma tabela com uma chave estrangeira para a entidade geral (ou generalizada, pai)



Pessoa	
nome	cpf
João	111.222...
Josefa	333.444...
Carlos	555.666...

Funcionario		
<u>id</u>	salario	cpf
1	1320	111.222...
2	2500	333.444...

Cliente		
<u>id</u>	saldo_devedor	cpf
1	-120.05	555.666...

# Exercício - Relacionamentos

---

- Construa um modelo conceitual para uma padaria, traduza-o para o modelo lógico no qual:
  - Funcionários possuem supervisores, que também são funcionários
  - Cada funcionário possui um cargo, com um salário associado ao cargo
  - Cada funcionário pode realizar uma venda
  - Cada venda é feita para um cliente
  - Clientes e funcionários são pessoas
  - Cada pessoa possui um CPF e um ou mais telefones

Descreva quais são as **chaves** (primárias e estrangeiras), **relacionamentos** (1:N ou N:N), seus **graus** (unário, binário, ternário) e **tabelas de junção**

# Referencial Bibliográfico

---

- KORTH, H.; SILBERSCHATZ, A.; SUDARSHAN, S. **Sistemas de bancos de dados**. 5. ed. Rio de Janeiro: Ed. Campus, 2006.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Ed. Campus, 2004. Tradução da 8ª edição americana.