



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

Modelo Lógico (Representação, Domínios de atributos e chaves)

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

Modelo Lógico

- 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			
<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

Representação com Diagrama de Esquemas

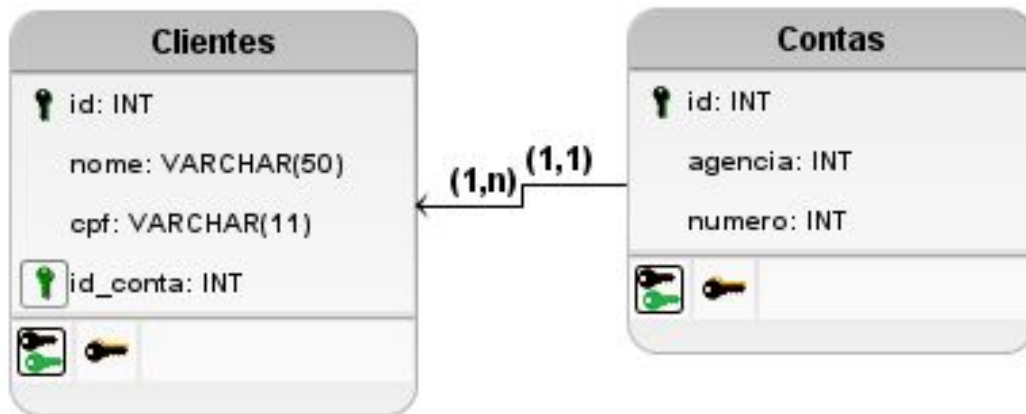
- Representar todos os atributos e instâncias em uma tabela ocupa muito espaço
 - **Solução:** descrever apenas o esquema da tabela usando o **diagrama de esquemas**

DEPARTAMENTO

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

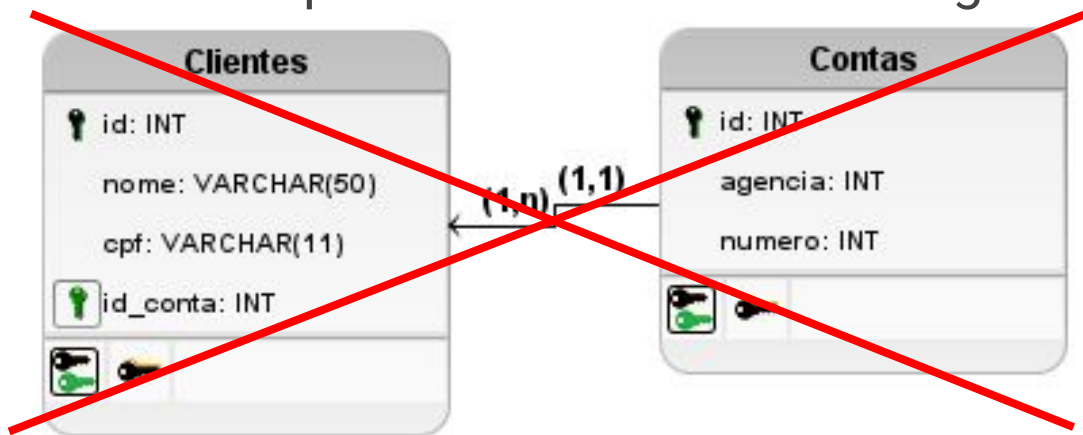
Representação semelhante à UML

- 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:** `Clientes(id, nome, cpf, id_conta)`
`Contas(id, agencia, 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:** $\text{Contas}(\text{id}, \text{agencia}, \text{numero})$
 - 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 os dados armazenados em nessa relação, em um determinado instante de tempo
- **r(R) = {t₁, t₂, ..., t_m}** onde **t_k = <v₁, v₂, ..., v_n>** é uma tupla
 - **v_i** são os valores de cada atributo **A_i** da tupla **t**
 - **v_i ∈ dom(A_i)**
- **Ex:** Seja R = Clientes(id, nome, cpf, id_conta)
 - t₁ = <1, 'Julia', 111.222.333-44, 2>
 - t₂ = <2, 'Carlos', 555.666.777-88, 1>
 - t₃ = <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:
 - $[0, 2^{32}] = [0, 4294967295]$

INT ou INTEGER

- 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 infinitos com exatidão, pois eles são sistemas binários
 - Porém números fracionários são infinitos
- **Problema:** ao fazer operações com números fracionários, podemos ter erros de aproximação
 - **Solução:** minimizar erros de aproximação definindo uma precisão para o número fracionário
 - Precisão simples (*FLOAT*) ou dupla (*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 true/false ou 1/0
 - **Ex:** Aprovação ou reprovação em uma disciplina

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

DATE e TIME

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

DATETIME

- **DATETIME:** Combinação de **DATE** e **TIME**, no formato 'AAAA-MM-DD HH:MM:SS'
 - A faixa suportada é entre '1000-01-01 00:00:00' e '9999-12-31 23:59:59'
 - **Ex:** armazenar o dia e horário que um funcionário chegou para trabalhar

VARCHAR, TEXT e LONGTEXT

- **VARCHAR (M)**: Uma string de tamanho variável de 1 a **M** caracteres
 - 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
- **LONGTEXT**: String com máximo de 4,294,967,295 caracteres

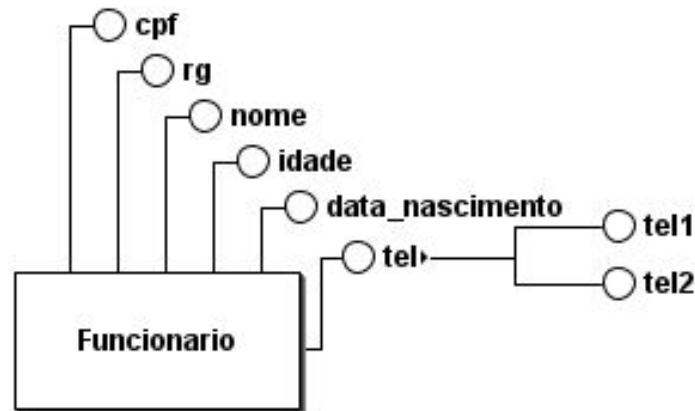
Chave em Banco de Dados

- 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

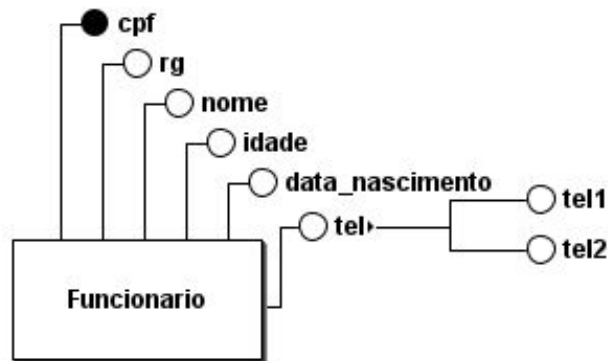
- São **atributos determinantes** que PODEM identificar registros de uma tabela de forma exclusiva
 - Não pode ter valores repetidos ou nulos
(restrição de integridade de entidade)

- Funcionários diferentes possuem CPFs e RGs distintos
- CPF e RG identificam cada funcionário de forma exclusiva



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
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

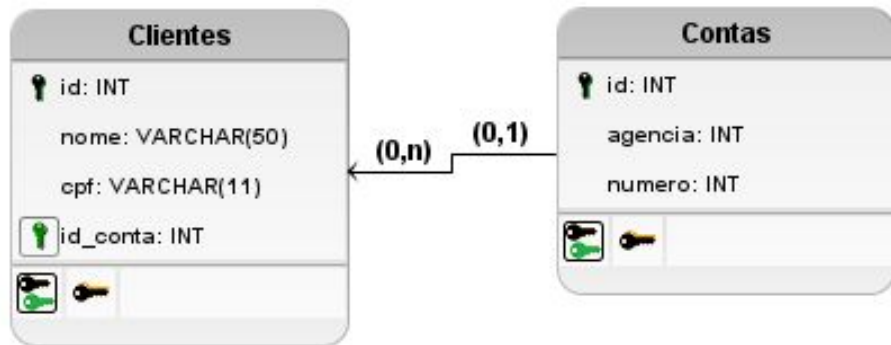
Tabela de Clientes				Tabela de Contas		
<u>ID</u>	<u>Nome</u>	<u>CPF</u>	<u>ID Conta</u>	<u>ID</u>	<u>Agência</u>	<u>Número</u>
1	Julia	111.222.333-44	2	1	3460	71542
2	Carlos	555.666.777-88	1	2	5421	65321
3	Amanda	123.456.789-00	3	3	7410	02145

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

Tabela de Clientes			FK	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

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



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

O atributo “*Cliente.id_conta*” é
uma chave estrangeira que
referencia o atributo
“*Contas.id*”

Também podemos dizer 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, 35, São Paulo, SP	M	55.000	NULL	1

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

Chaves estrangeiras podem conter o valor NULL

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 Timpira, 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

Exercício - Chaves

- Escolha um dos sistemas descritos no link abaixo:
 - https://github.com/andre-romano/tutorial_php/tree/master/projetos
- Quais são as chaves primárias e estrangeiras das tabelas do sistema?
- Porque não podemos ter valores NULL nas chaves primárias?

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.