

# BASES DE DADOS

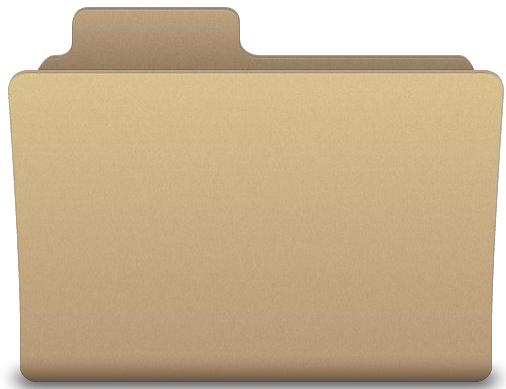
---



# Déc. 50

DADOS PESSOAIS					
Nome Completo Comprador					
Carteira de Identidade	Orgão Carteira Profissional	Série	Outro documento		
CPF	Nacionalidade	Natural de	Data de Nascimento		
Filiação	Pai	Mãe			
Endereço Residencial			Nº	Apto.	
Bairro	Cidade	Estado	Telefone		
CEP	Casa Própria/Augada/Com os pais/ Hotel/República/			Há quanto tempo	
Empresa que trabalha			Seção / Sala		
Endereço			Nº	Telefone	
Bairro	Cidade	Estado	Data admissão	Renda Mensal	Cargo
Estado Civil	Nome do Esposo (a)		Profissão	Renda Mensal	
REFERÉNCIAS COMERCIAIS					
Referência	Referência				
Referência	Referência				
Pessoa Conhecida					
Nome					
Endereço			Telefone		

DADOS PESSOAIS				
Nome Completo		Comprador		
Cartera de Identidade	Orgão Cartera	Profissional	Série	Outro documento
CPF	Nacionalidade	Natural de	Data de Nascimento	
Filiação	Tia	Iúdea		
Endereço Residencial		Nº	Apto.	
Bairro	Cidade	Estado	Telefone	
CEP	Casa Própria/Augada/Com os pais/ Hotel/República/ Há quanto tempo			
Empresa que trabalha		Seção / Sala		
Endereço		Nº	Telefone	
Bairro	Cidade	Estado	Data admissão	Renda Mensal
Estado Civil	Nome do Espouse (s)		Profissão	Renda Mensal
REFERENCIAS COMERCIAIS				
Referência	Referência			
Referência	Referência			
Pessoas Conhecidas				
Name				
Endereço	Telefone			



DADOS PESSOAIS		
Nome Completo:	Completo	
Código de Identidade:	Socorro	Nº documento:
Profissão:		
CPF:	123.456.789-00	Natural de:
Estado:	SP	Data de Nascimento:
Endereço Residencial:		
Sexo:		
Estado Civil:		
Nome do Espouse (o) / Parente:		Pessoal
REFERENCIAS COMERCIAIS		
Patrocinador:		Referência
Patrocinado:		Referência
Passos Concedida:		
Endereço:		
Telefone:		

Registos

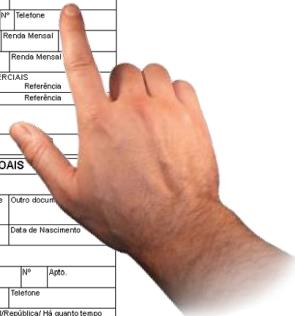


Tabelas



Ficheiros





**Arquivos Sequenciais**

CEP	Casa Própria/Augada/Com os pais/Hotel/República/ Há quanto tempo			
Empresa que trabalha				
Endereço				
Bairro	Cidade	Estado	Data admissão	Renda Mensal
Estado Civil Nome do Espaco (e) Profissão Renda Mensal				
REFERENCIAS COMERCIAIS				
Referência	Referência			
Referência	Referência			
Pessoas Conhecida	Nome			
Endereço				
DADOS PESSOAIS				
Nome Completo		Comprador		
Carteira de Identidade	Orgão Cartera e Profissional	Série	Outro documento	
CPF	Nacionalidade	Natural de	Data de Nascimento	
Filiação	Pai	Mãe		
Endereço Residencial				
Bairro	Cidade	Estado	Telefone	Nº Apt.
CEP	Casa Própria/Augada/Com os pais/ Hotel/República/ Há quanto tempo			
Empresa que trabalha				
Endereço				
Bairro	Cidade	Estado	Data admissão	Renda Mensal
Estado Civil	Nome do Espaco (e)	Profissão	Renda Mensal	
REFERENCIAS COMERCIAIS				
Referência	Referência			
Referência	Referência			
Pessoas Conhecida	Nome			
Endereço				
DADOS PESSOAIS				
Nome Completo		Comprador		
Carteira de Identidade	Orgão Cartera e Profissional	Série	Outro documento	
CPF	Nacionalidade	Natural de	Data de Nascimento	
Filiação	Pai	Mãe		
Endereço Residencial				
Bairro	Cidade	Estado	Telefone	Nº Apt.
CEP	Casa Própria/Augada/Com os pais/ Hotel/República/ Há quanto tempo			
Empresa que trabalha				
Endereço				
Bairro	Cidade	Estado	Data admissão	Renda Mensal
Estado Civil	Nome do Espaco (e)	Profissão	Renda Mensal	
REFERENCIAS COMERCIAIS				
Referência	Referência			
Referência	Referência			
Pessoas Conhecida	Nome			
Endereço				



DADOS PESSOAIS			
Nome Completo	Completo	Celular	Outro documento
Carolina de	Carolina	Série	Data de documento
CEP	Nacionalidade	Natural de	Data de Nascimento
Nome	Nome	Nome	Nome
Endereço Residencial	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Endereço que trabalha	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome da Esposa (O) / Prófissional			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
REFERENCIAS COMERCIAIS			
Nome	Nome	Nome	Nome
Endereço	Endereço	Endereço	Endereço
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Referência			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP

DADOS PESSOAIS			
Nome Completo	Completo	Celular	Outro documento
Carolina de	Carolina	Série	Data de documento
CEP	Nacionalidade	Natural de	Data de Nascimento
Nome	Nome	Nome	Nome
Endereço Residencial	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Endereço que trabalha	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome da Esposa (O) / Prófissional			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
REFERENCIAS COMERCIAIS			
Nome	Nome	Nome	Nome
Endereço	Endereço	Endereço	Endereço
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Referência			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP

DADOS PESSOAIS			
Nome Completo	Completo	Celular	Outro documento
Carolina de	Carolina	Série	Data de documento
CEP	Nacionalidade	Natural de	Data de Nascimento
Nome	Nome	Nome	Nome
Endereço Residencial	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Endereço que trabalha	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome da Esposa (O) / Prófissional			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
REFERENCIAS COMERCIAIS			
Nome	Nome	Nome	Nome
Endereço	Endereço	Endereço	Endereço
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Referência			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP

DADOS PESSOAIS			
Nome Completo	Completo	Celular	Outro documento
Carolina de	Carolina	Série	Data de documento
CEP	Nacionalidade	Natural de	Data de Nascimento
Nome	Nome	Nome	Nome
Endereço Residencial	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Endereço que trabalha	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome da Esposa (O) / Prófissional			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
REFERENCIAS COMERCIAIS			
Nome	Nome	Nome	Nome
Endereço	Endereço	Endereço	Endereço
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Referência			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP

DADOS PESSOAIS			
Nome Completo	Completo	Celular	Outro documento
Carolina de	Carolina	Série	Data de documento
CEP	Nacionalidade	Natural de	Data de Nascimento
Nome	Nome	Nome	Nome
Endereço Residencial	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Endereço que trabalha	Endereço	Endereço	Endereço
Nome	Cidade	UF	Estado
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome da Esposa (O) / Prófissional			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
REFERENCIAS COMERCIAIS			
Nome	Nome	Nome	Nome
Endereço	Endereço	Endereço	Endereço
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP
Referência			
Nome	Nome	Nome	Nome
CEP	CEP	CEP	CEP

1  
2  
3  
4

5  
6  
7

8  
9

Acesso Direto





Déc. 60



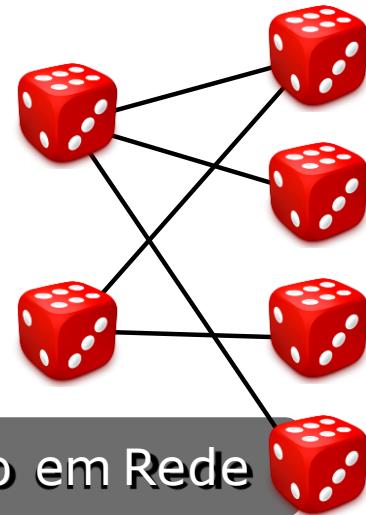
CODASYL  
**COBOL**

Base de Dados

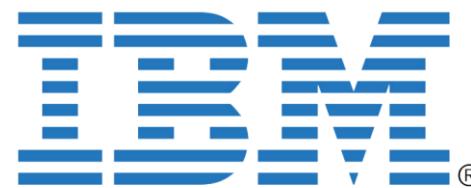




Modelo Hierárquico



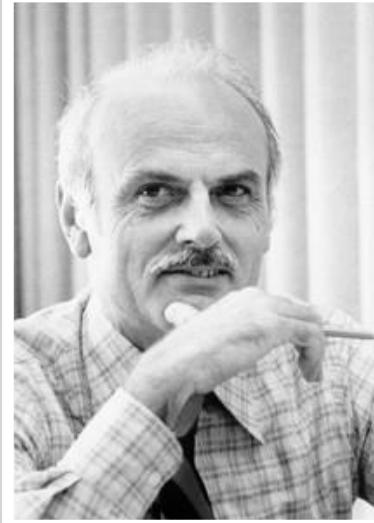
Modelo em Rede



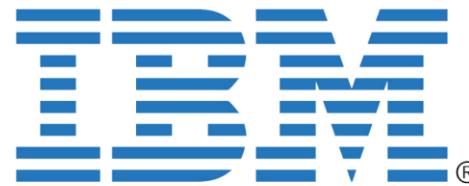
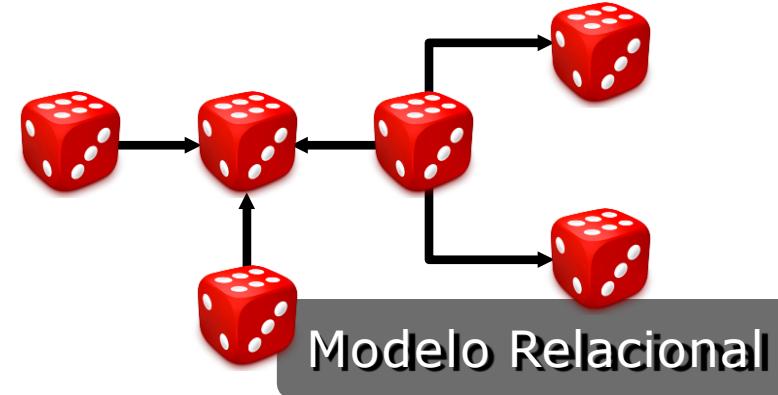




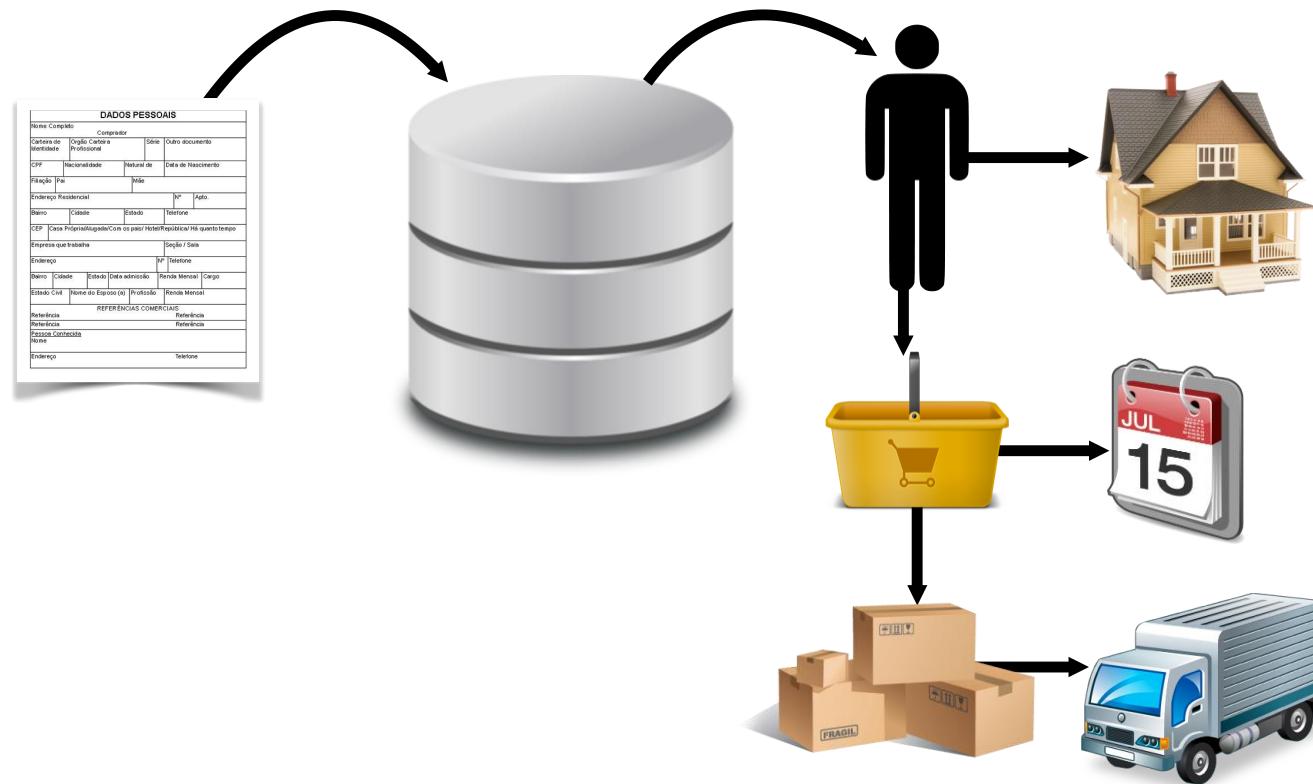
Déc. 70



Edgar F. Codd



# O modelo relacional permite..



através de uma linguagem de exploração!

# SEQUEL



**Structured English Query Language**  
**SEQUEL**

SQL



Structured Query Language  
**SQL**





# Em 1994 na Suécia

**GPL**



**MySQL®**

 MariaDB

The logo for MariaDB features a stylized orange seal or sea lion leaping to the left, positioned next to the word "MariaDB" in a lowercase, sans-serif font.

Michael Widenius



David Axmark

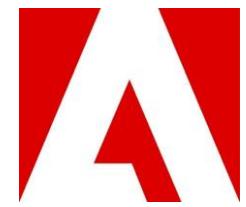
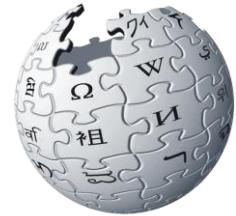
 **Sun**  
microsystems

The Sun Microsystems logo consists of a blue diamond shape composed of several smaller triangles, followed by the word "Sun" in a large, blue, italicized serif font, and "microsystems" in a smaller, blue, sans-serif font below it.

**ORACLE®**



Google



..|..|..|..  
cisco™

ebay™





Remover o mariaDB

# Alguns notas:

SGBD phpMyAdmin

Utilizador por defeito: root

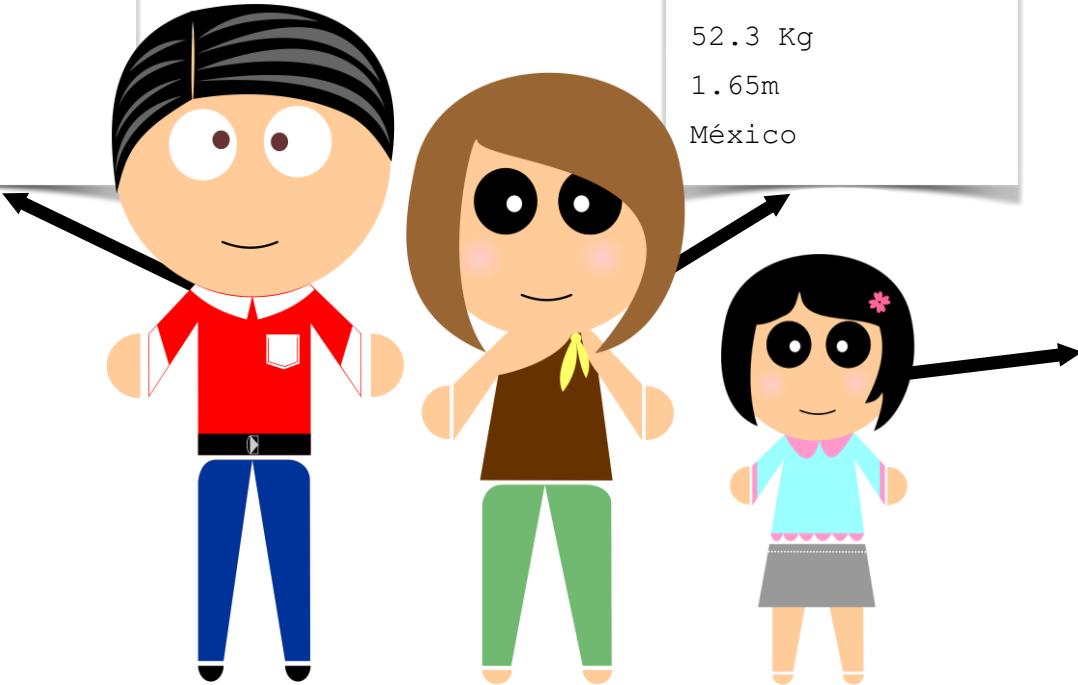
Password:

Status ( só funciona na consola )

```
use nome_da_basededados;  
describe nome_da_tabela;
```

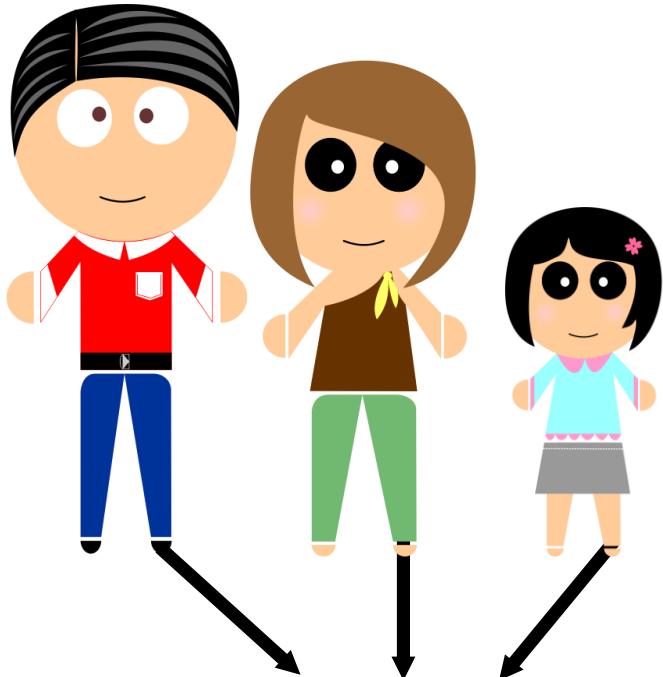
```
show databases;  
show tables;
```

Godofredo  
32 anos  
Masc  
78.5 Kg  
1.83 m  
Portugal

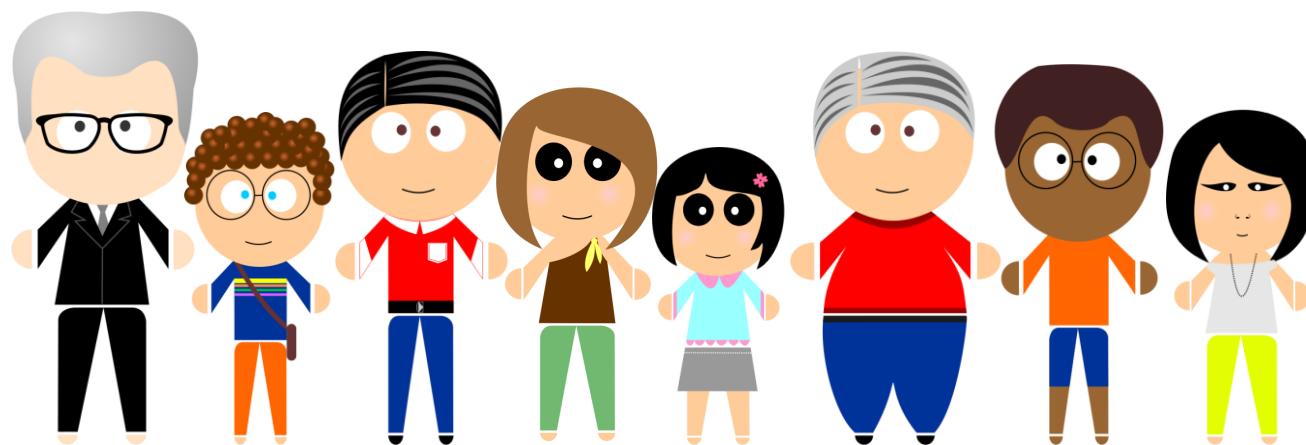


Dolores  
30 anos  
Fem  
52.3 Kg  
1.65m  
México

Godolores  
3 anos  
Fem  
25.8 Kg  
0.89m  
EUA



nome  
idade  
sexo  
peso  
altura  
nacionalidade





nome  
idade  
sexo  
peso  
altura  
nacionalidade







Base de  
dados



**CREATE DATABASE minhaBasedeDados;**

# Comentários

Comentários de uma linha #

Comentários de várias linhas /\* isto é um comentário  
de várias linhas\*/



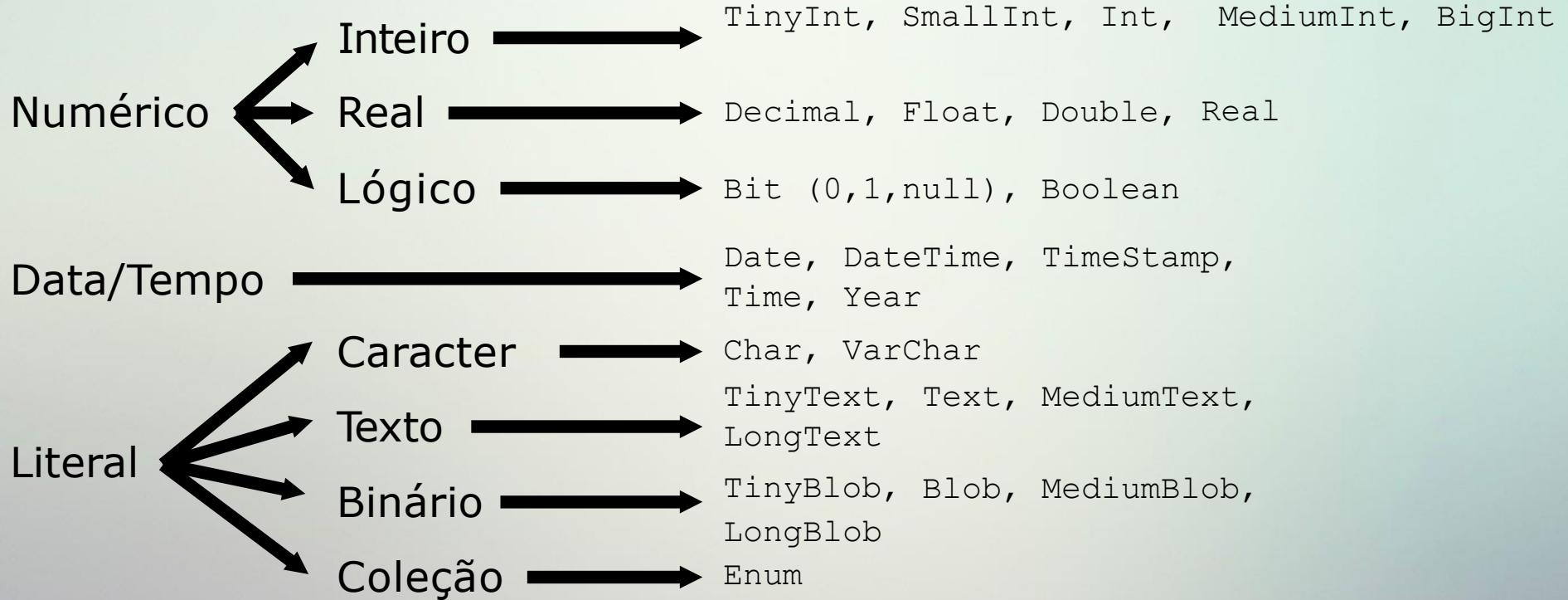
nome  
idade  
sexo  
peso  
altura  
nacionalidade

*Tabela*  
*Campos*



```
CREATE TABLE pessoas (
    nome ,
    idade ,
    sexo ,
    peso ,
    altura ,
    nacionalidade
);
```

# Tipos Primitivos





nome  
idade  
sexo  
peso  
altura  
nacionalidade

```
CREATE TABLE pessoas (
    nome varchar(30),
    idade tinyint,
    sexo char(1),
    peso float,
    altura float,
    nacionalidade varchar(20)
);
```

# Vamos melhorar a nossa bd!

Drop database minhaprimeiradb; #apagar a bd atual

```
CREATE DATABASE minhaprimeiradb  
DEFAULT CHARACTER SET utf8mb4 #constraints  
DEFAULT COLLATE utf8mb4_bin #constraints;
```

# Criar uma tabela com constraints

```
CREATE TABLE pessoas (
    id int NOT NULL AUTO_INCREMENT, #constraint
    nome varchar(30) NOT NULL, #constraint NOT NULL
    nascimento date,
    sexo enum('M', 'F'),
    peso decimal(5,2), #5 números 2 decimais
    altura decimal(3,2),
    nacionalidade varchar(20) DEFAULT 'Portugal', #constraint
    PRIMARY KEY (id)
) DEFAULT CHARSET = utf8mb4;
```

# Inserir dados numa tabela

```
INSERT INTO pessoas  
(nome, nascimento, sexo, peso, altura) #falta nacionalidade e id  
VALUES  
('Godofredo', '1984-01-02', 'M', '78.5', '1.83');
```

```
INSERT INTO pessoas VALUES  
(DEFAULT, 'Godofredo', '1984-01-02', 'M', '78.5', '1.83', 'Portugal');
```

# Inserir múltiplos dados numa tabela

```
INSERT INTO pessoas  
(id, nome, nascimento, sexo, peso, altura, nacionalidade)  
VALUES  
(DEFAULT, 'Ana', '1975-12-22', 'F', '52.3', '1.45', 'EUA'),  
(DEFAULT, 'Pedro', '2000-07-15', 'M', '52.3', '1.45', 'Brasil'),  
(DEFAULT, 'Maria', '1999-05-30', 'F', '75.9', '1.70', 'Portugal');
```

## Adicionar uma nova Coluna

```
ALTER TABLE pessoas  
ADD COLUMN profissao varchar(10);  
#vai adicionar o campo no final da tabela  
#o campo column é facultativo
```

## Remover uma Coluna

```
ALTER TABLE pessoas  
DROP COLUMN profissao;
```

## Adicionar uma nova Coluna depois de...

```
ALTER TABLE pessoas  
ADD COLUMN profissao varchar(10) after nome;
```

```
ALTER TABLE pessoas  
ADD COLUMN codigo int FIRST;
```

## Modificar Definições

```
ALTER TABLE pessoas  
MODIFY COLUMN profissao varchar(20) not null;
```

## Renomear uma coluna

```
ALTER TABLE pessoas  
CHANGE COLUMN profissao prof varchar(20);
```

# Renomear o nome de uma tabela

```
ALTER TABLE pessoas  
RENAME TO novas_pessoas;
```

# Mais uma Tabela...

```
CREATE TABLE IF NOT EXISTS cursos (
    nome varchar(30) NOT NULL UNIQUE,
    descricao text,
    carga int UNSIGNED,
    totaulas int,
    ano year DEFAULT '2016'
) DEFAULT CHARSET = utf8mb4;
```

# Alterar a tabela para...

## Adicionar Chave Primária

```
ALTER TABLE cursos  
ADD COLUMN idcurso int FIRST;
```

```
ALTER TABLE cursos  
ADD PRIMARY KEY(idcurso);
```

Ou

```
ALTER TABLE cursos  
ADD COLUMN idcurso int FIRST,  
ADD PRIMARY KEY(idcurso);
```

# Apagar a Tabela

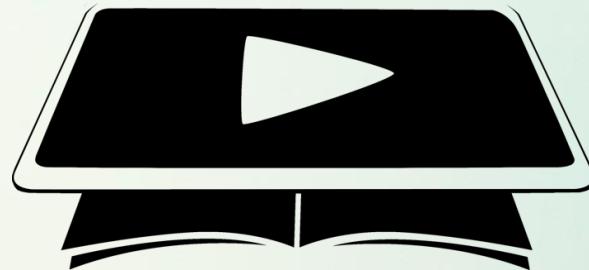
```
DROP TABLE IF EXISTS cursos;
```

# Tabelas atuais - Alterar os registos



**pessoas**

id  
nome  
profissao  
nascimento  
sexo  
peso  
altura  
nacionalidade



**cursos**

idcurso  
nome  
descricao  
carga  
totaulas  
ano

# Alterar os registos da Tabela ‘cursos’

```
INSERT INTO cursos VALUES  
('1', 'HTML4', 'Curso de HTML5', '40', '37', '2014'),  
('2', 'Algoritmos', 'Lógica Programação', '20', '15', '2014'),  
('3', 'Photoshop', 'Dicas de Photoshop CC', '10', '8', '2014'),  
('4', 'PGP', 'Curso PHP para iniciantes', '40', '20', '2010'),  
('5', 'Jarva', 'Introdução Linguagem Java', '10', '29', '2000'),  
('6', 'MySQL', 'Banco de Dados MySQL', '30', '15', '2016'),  
('7', 'Word', 'Curso completo de Word', '40', '30', '2016'),  
('8', 'Sapateado', 'Danças Rítmicas', '40', '30', '2018'),  
('9', 'Cozinha Árabe', 'Aprenda a fazer', '40', '30', '2018'),  
('10', 'Youtuber', 'Gerar polêmica', '5', '2', '2018');
```

# Alterar os registos da Tabela ‘cursos’

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>carga</b>	<b>totaulas</b>	<b>ano</b>
1	<b>HTML4</b>	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	<b>PGP</b>	...	40	20	<b>2010</b>
5	<b>Jarva</b>	...	<b>10</b>	29	<b>2000</b>
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Corrigir Linhas Incorretas

```
UPDATE cursos  
SET nome = 'HTML5'  
WHERE idcurso = '1';
```

```
UPDATE cursos  
SET nome = 'PHP', ano = '2015'  
WHERE idcurso = '4';
```

```
UPDATE cursos  
SET nome = 'Java', carga = '40', ano = '2015'  
WHERE idcurso = '5';
```

idcurso	nome	carga	ano
1	HTML4	40	2014
4	PGP	40	2010
5	Jarva	10	2000

## Remover uma Linha

```
DELETE FROM cursos  
WHERE idcurso='8';
```

```
DELETE FROM cursos  
WHERE ano='2018'  
LIMIT 3; #só afeta uma linha
```

## Remover todas as Linha

```
TRUNCATE TABLE cursos;  
DELETE FROM cursos;
```

## *DDL*

*Data Definition  
Language*

CREATE DATABASE  
CREATE TABLE  
DROP DATABASE  
DROP TABLE  
ALTER TABLE

## *DML*

*Data Manipulation  
Language*

INSERT INTO  
UPDATE tabela  
DELETE tabela  
TRUNCATE tabela

# Registros da Tabela ‘cursos’

```
SELECT *
FROM cursos;
```

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>carga</b>	<b>totaulas</b>	<b>ano</b>
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Ordenação Crescente

```
SELECT *
FROM cursos
ORDER BY nome;
```

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>carga</b>	<b>totaulas</b>	<b>ano</b>
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Ordenação Decrescente

```
SELECT *
FROM cursos
ORDER BY nome DESC;
```

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>totaulas</b>	<b>carga</b>	<b>ano</b>
10	YouTuber	...	5	2	2018
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
4	PHP	...	40	20	2015
3	Photoshop	...	10	8	2014
6	MySQL	...	30	15	2016
5	Java	...	40	29	2015
1	HTML5	...	40	37	2014
9	Cozinha Árabe	...	40	30	2018
2	Algoritmos	...	20	15	2014

# Selecionar Colunas

```
SELECT nome, carga, ano  
FROM cursos  
ORDER BY nome;
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Selecionar Linhas

```
SELECT *
FROM cursos
WHERE ano = '2016'
ORDER BY nome;
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Selecionar Linhas e Colunas

```
SELECT nome, descricao, carga  
FROM cursos  
WHERE ano = '2016'  
ORDER BY nome;
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Operadores relacionais

Igual	Diferente	Maior que	Menor que	Maior ou igual	Menor ou igual
=	< > ou !=	>	<	>=	<=

# Selecionar Linhas e Colunas

```
SELECT nome, descricao  
FROM cursos  
WHERE ano <= '2015'  
ORDER BY nome, ano;
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Selecionar Intervalos

```
SELECT *
FROM cursos
WHERE totaulas BETWEEN '20' AND '30'
ORDER BY nome;
#é inclusivo o 20 e o 30
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Selecionar Valores

```
SELECT idcurso, nome  
FROM cursos  
WHERE ano IN ('2014', '2016', '2018')  
ORDER BY nome;
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Combinar Testes

```
SELECT *
FROM cursos
WHERE carga > 35 AND totaulas < 30
ORDER BY nome;
```

idcurso	nome	descricao	carga	totaulas	ano
2	Algoritmos	...	20	15	2014
9	Cozinha Árabe	...	40	30	2018
1	HTML5	...	40	37	2014
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
8	Sapateado	...	40	30	2018
7	Word	...	40	30	2016
10	YouTuber	...	5	2	2018

# Operadores Lógicos

p	q	p AND q
V	V	V
V	F	F
F	V	F
F	F	F

p	q	p OR q
V	V	V
V	F	V
F	V	V
F	F	F

q	not q
V	F
F	V

# Seleção por nome

```
SELECT *
FROM cursos
WHERE nome = 'PHP';
```

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>carga</b>	<b>totaulas</b>	<b>ano</b>
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Usar o Operador LIKE

```
SELECT *
FROM cursos
WHERE nome LIKE 'P%';
```

Parecido

Nenhum ou  
Vários Caracteres

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Wildcards

```
SELECT * FROM cursos  
WHERE nome LIKE '%A';
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Wildcards

```
SELECT * FROM cursos  
WHERE nome LIKE '%A%';
```

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>carga</b>	<b>totaulas</b>	<b>ano</b>
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Wildcards

```
SELECT * FROM cursos  
WHERE nome LIKE 'PH%P';
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Wildcards

```
SELECT * FROM cursos  
WHERE nome LIKE 'P%H_';
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Distinto

```
SELECT DISTINCT carga  
FROM cursos;
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Funções de Agregação

```
SELECT COUNT (*)
FROM cursos;
```

10					
idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Funções de Agregação

```
SELECT COUNT(*)
FROM cursos
WHERE nome LIKE '%P%';
```

3					
idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Funções de Agregação

```
SELECT MAX(totaulas)  
FROM cursos;
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Funções de Agregação

```
SELECT MIN(totaulas)  
FROM cursos;
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Funções de Agregação

```
SELECT SUM(totaulas)  
FROM cursos;
```

<b>idcurso</b>	<b>nome</b>	<b>descricao</b>	<b>carga</b>	<b>totaulas</b>	<b>ano</b>
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Funções de Agregação

```
SELECT AVG(totaulas)  
FROM cursos;
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

```
SELECT ROUND(AVG(totaulas), 2)  
FROM cursos;
```



Exercícios...

“Uma listagem com o nome  
de todas as pessoas”



“Uma listagem com os  
dados de todos os  
utilizadores que nasceram  
entre 1/Jan/2000 e  
31/Dez/2015”



“Uma listagem com o nome e  
a profissão de todos os  
homens que trabalham como  
Programadores”



“Uma listagem com os dados de todas as mulheres que nasceram em Portugal e que tem o nome iniciado com a letra J”



“Uma listagem com o nome e nacionalidade de todos os homens que têm Silva no nome, não nasceram em Portugal e pesam menos de 100 Kg”



**“Qual é a maior altura das pessoas do sexo feminino que moram em Portugal?”**



**“Qual é a média de peso  
dos utilizadores do EUA?”**



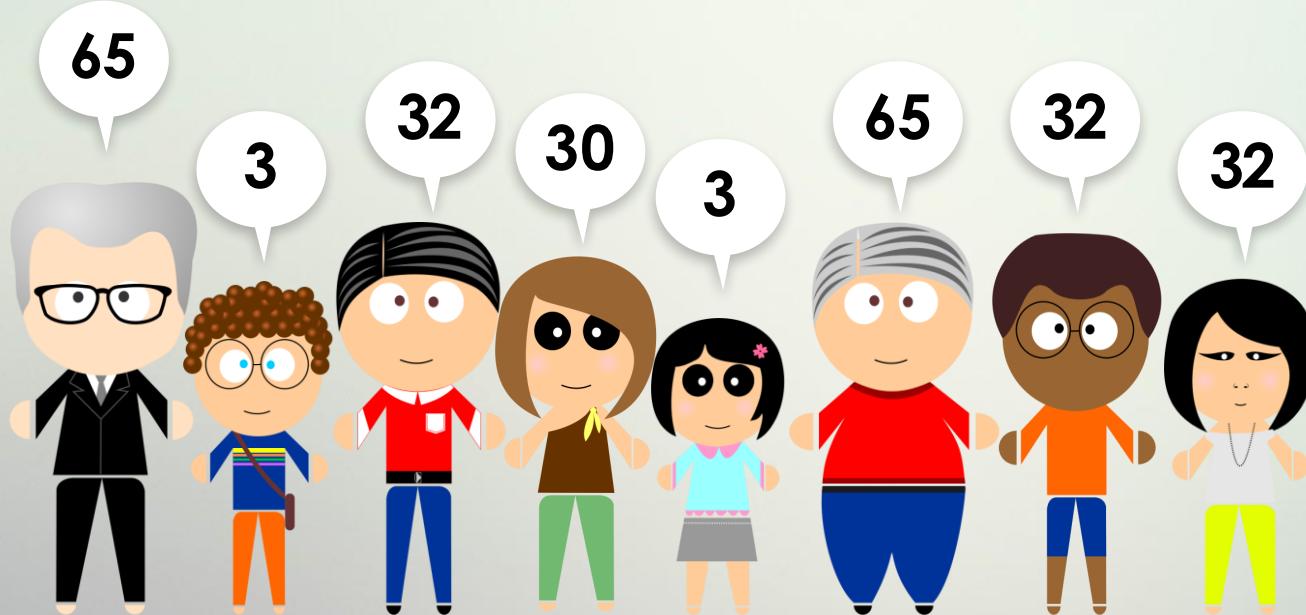
“Qual é o menor peso entre os utilizadores mulheres que nasceram fora de Portugal e entre 01/Jan/1990 e 31/Dez/2000?”



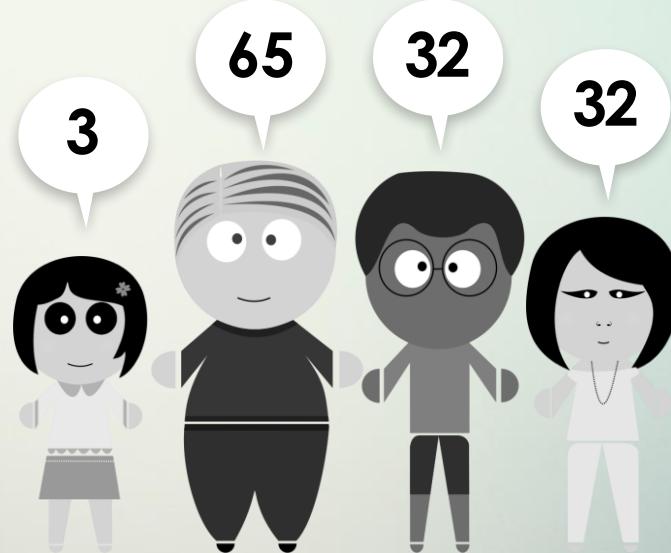
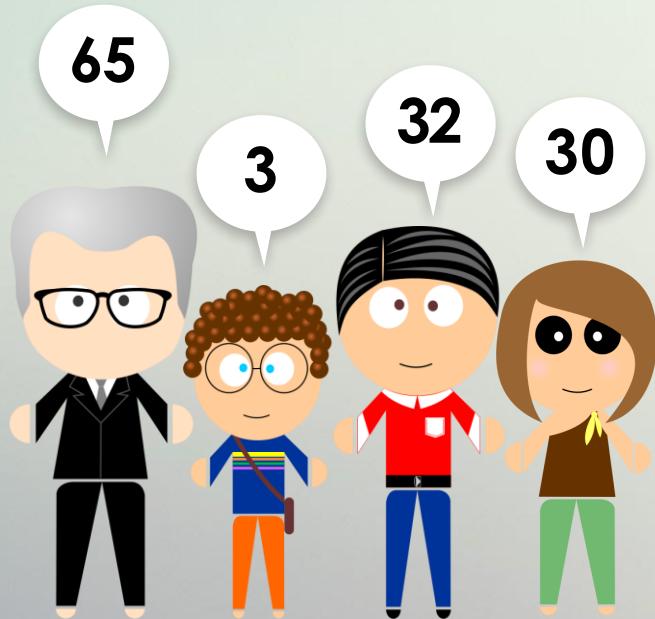
**“Quantos utilizadores  
mulheres têm mais de  
1.90m de altura?”**



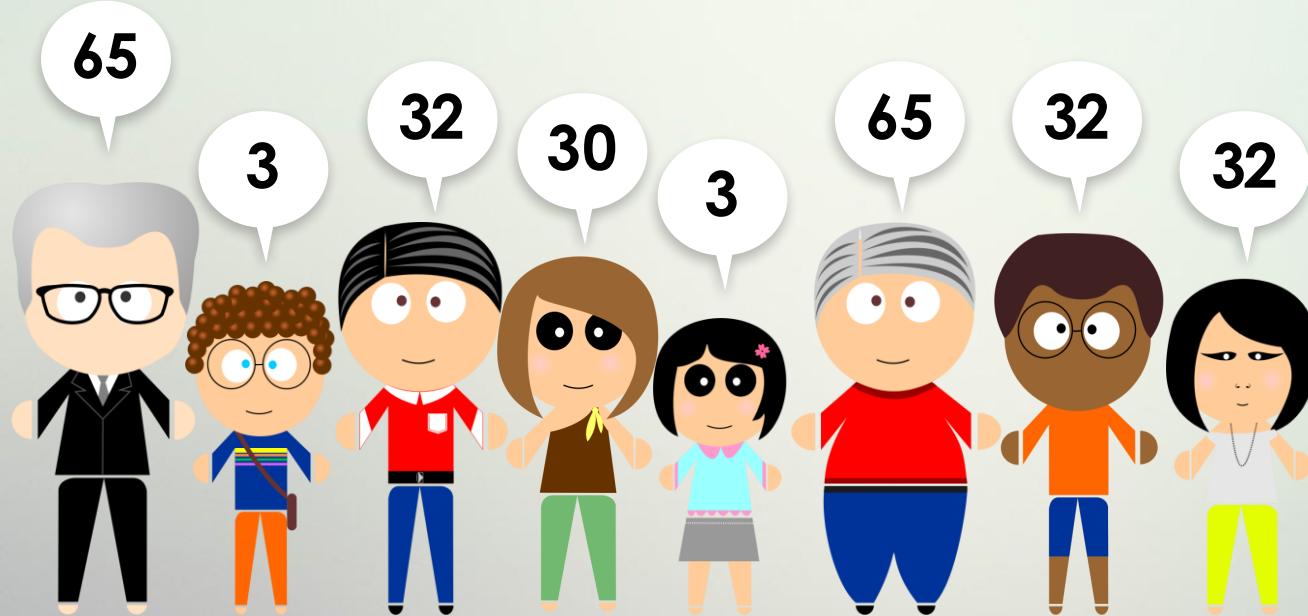
# Distinto



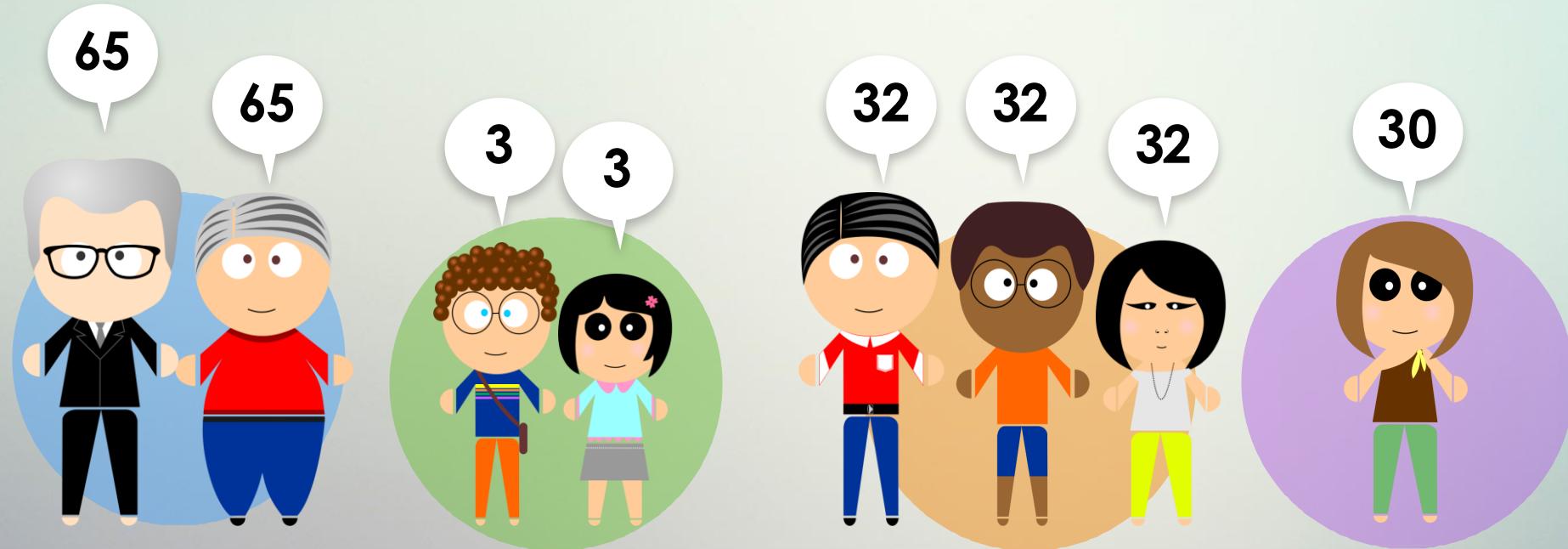
# Distinto



# Agrupar



# Agrupar



# Distinto

```
SELECT DISTINCT carga  
FROM cursos  
ORDER BY carga;
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

# Distinto

```
SELECT DISTINCT carga  
FROM cursos  
ORDER BY carga;
```

idcurso	nome	descricao	carga	totaulas	ano	carga
1	HTML5	...	40	37	2014	5
2	Algoritmos	...	20	15	2014	10
3	Photoshop	...	10	8	2014	20
4	PHP	...	40	20	2015	30
5	Java	...	40	29	2015	40
6	MySQL	...	30	15	2016	
7	Word	...	40	30	2016	
8	Sapateado	...	40	30	2018	
9	Cozinha Árabe	...	40	30	2018	
10	YouTuber	...	5	2	2018	

# Agrupar Registros

```
SELECT carga  
FROM cursos  
GROUP BY carga;
```

*Agrupado*

idcurso	nome	descricao	carga	totaulas	ano	carga
1	HTML5	...	40	37	2014	40
2	Algoritmos	...	20	15	2014	20
3	Photoshop	...	10	8	2014	10
4	PHP	...	40	20	2015	30
5	Java	...	40	29	2015	5
6	MySQL	...	30	15	2016	
7	Word	...	40	30	2016	
8	Sapateado	...	40	30	2018	
9	Cozinha Árabe	...	40	30	2018	
10	YouTuber	...	5	2	2018	

# Agrupar & Agregar

```
SELECT carga , COUNT(*)  
FROM cursos  
GROUP BY carga;
```

idcurso	nome	descricao	carga	totaulas	ano	carga	count()
1	HTML5	...	40	37	2014	40	6
2	Algoritmos	...	20	15	2014	20	1
3	Photoshop	...	10	8	2014	10	1
4	PHP	...	40	20	2015	30	1
5	Java	...	40	29	2015	5	1
6	MySQL	...	30	15	2016		
7	Word	...	40	30	2016		
8	Sapateado	...	40	30	2018		
9	Cozinha Árabe	...	40	30	2018		
10	YouTuber	...	5	2	2018		

# Agrupar & Agregar

```
SELECT carga , COUNT(*)  
FROM cursos  
WHERE totaulas > 10  
GROUP BY carga;
```

idcurso	nome	descricao	carga	totaulas	ano	carga	count()
1	HTML5	...	40	37	2014	40	6
2	Algoritmos	...	20	15	2014	20	1
3	Photoshop	...	10	8	2014	30	1
4	PHP	...	40	20	2015		
5	Java	...	40	29	2015		
6	MySQL	...	30	15	2016		
7	Word	...	40	30	2016		
8	Sapateado	...	40	30	2018		
9	Cozinha Árabe	...	40	30	2018		
10	YouTuber	...	5	2	2018		

# Agrupar & Agregar

```
SELECT carga ,COUNT(*)  
FROM cursos  
GROUP BY carga  
HAVING COUNT(nome) > 3;
```

idcurso	nome	descricao	carga	totaulas	ano
1	HTML5	...	40	37	2014
2	Algoritmos	...	20	15	2014
3	Photoshop	...	10	8	2014
4	PHP	...	40	20	2015
5	Java	...	40	29	2015
6	MySQL	...	30	15	2016
7	Word	...	40	30	2016
8	Sapateado	...	40	30	2018
9	Cozinha Árabe	...	40	30	2018
10	YouTuber	...	5	2	2018

carga	count()
40	6

# Agrupar & Agregar

```
SELECT ano , COUNT(*) as 'contador'  
FROM cursos  
GROUP BY ano  
ORDER BY (COUNT(*)) desc;
```

```
SELECT ano, COUNT(*) as Contador  
FROM cursos  
GROUP BY ano  
HAVING COUNT(ano) >= 5  
ORDER BY (COUNT(*));
```

```
SELECT ano , COUNT(*)  
FROM cursos  
WHERE totaulas >= 30  
GROUP BY ano  
HAVING ano > 2013  
ORDER BY (COUNT(*));
```

# Agrupar & Agregar

```
SELECT AVG(carga) FROM cursos;
```

```
SELECT carga ,COUNT(*)  
FROM cursos  
WHERE ano > 2015  
GROUP BY carga  
HAVING carga > 36.3;
```

```
SELECT carga ,COUNT(*)  
FROM cursos  
WHERE ano > 2015  
GROUP BY carga  
HAVING carga > (SELECT AVG(carga) FROM cursos);
```



Exercícios...

**“Uma lista com as profissões e as respetivas quantidades”**



**“Quantos utilizadores homens e quantas mulheres nasceram depois de 01/Jan/2005?”**



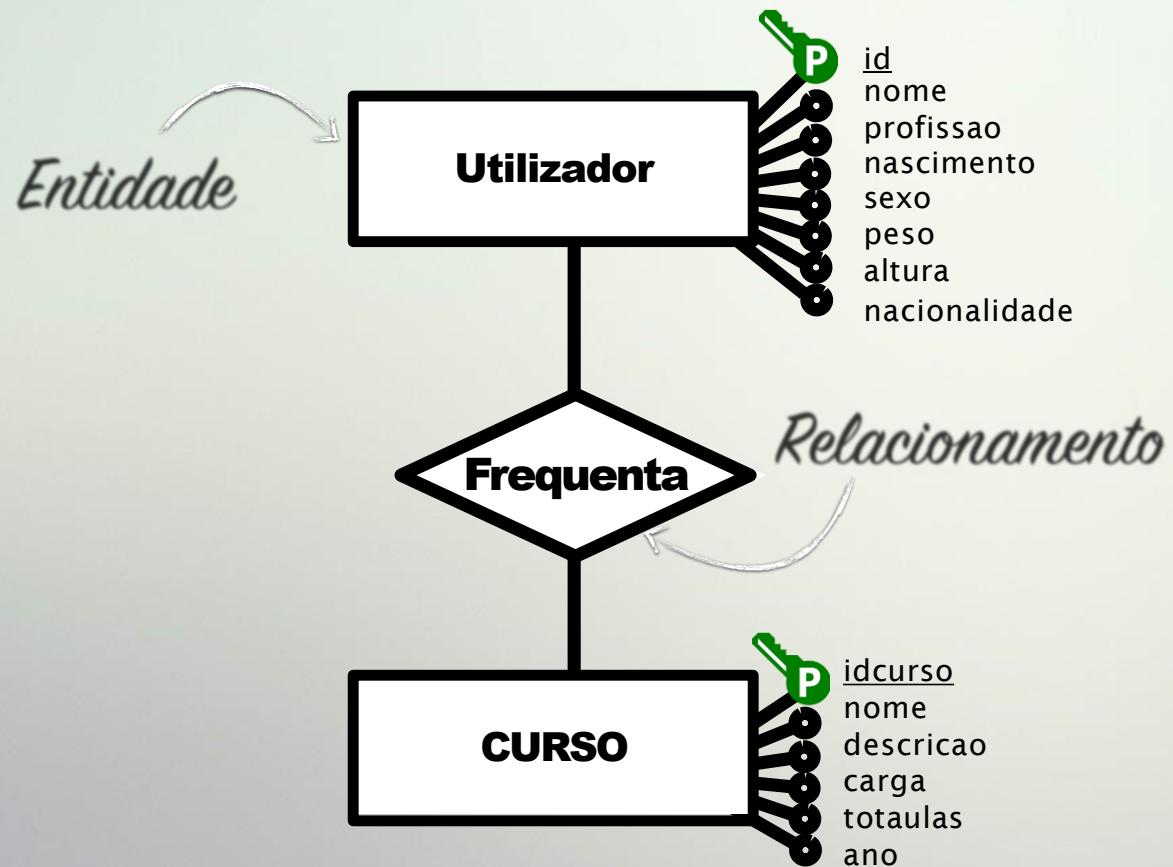
**“Uma listagem com o número de utilizadores e o país desses utilizadores que nasceram fora de Portugal. Só nos interessam os países com mais de 3 utilizadores com essa nacionalidade”**



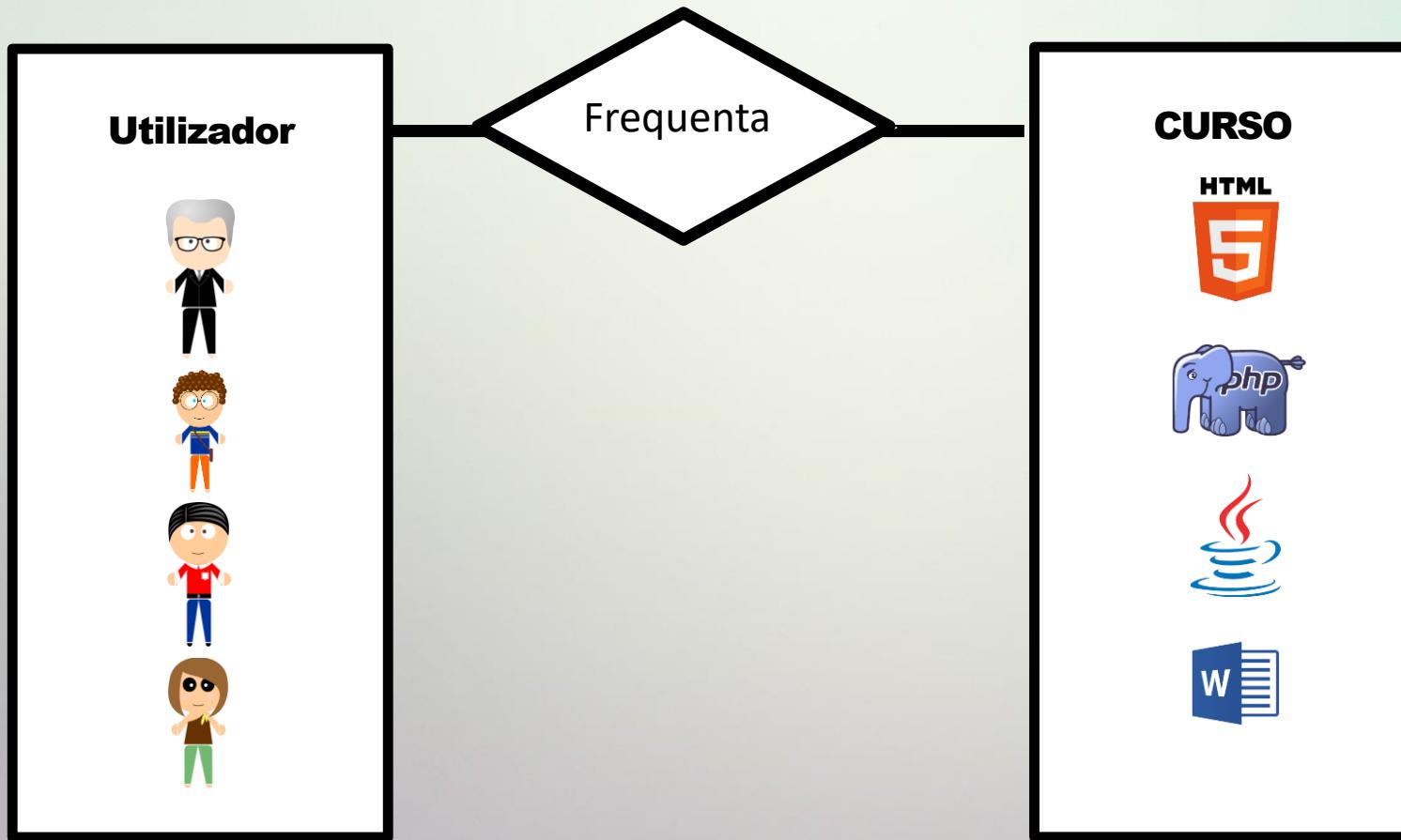
**“Uma listagem agrupada pela altura dos utilizadores, onde mostra quantas pessoas pesam mais de 100Kg e que estão acima da média de altura de todos os utilizadores”**



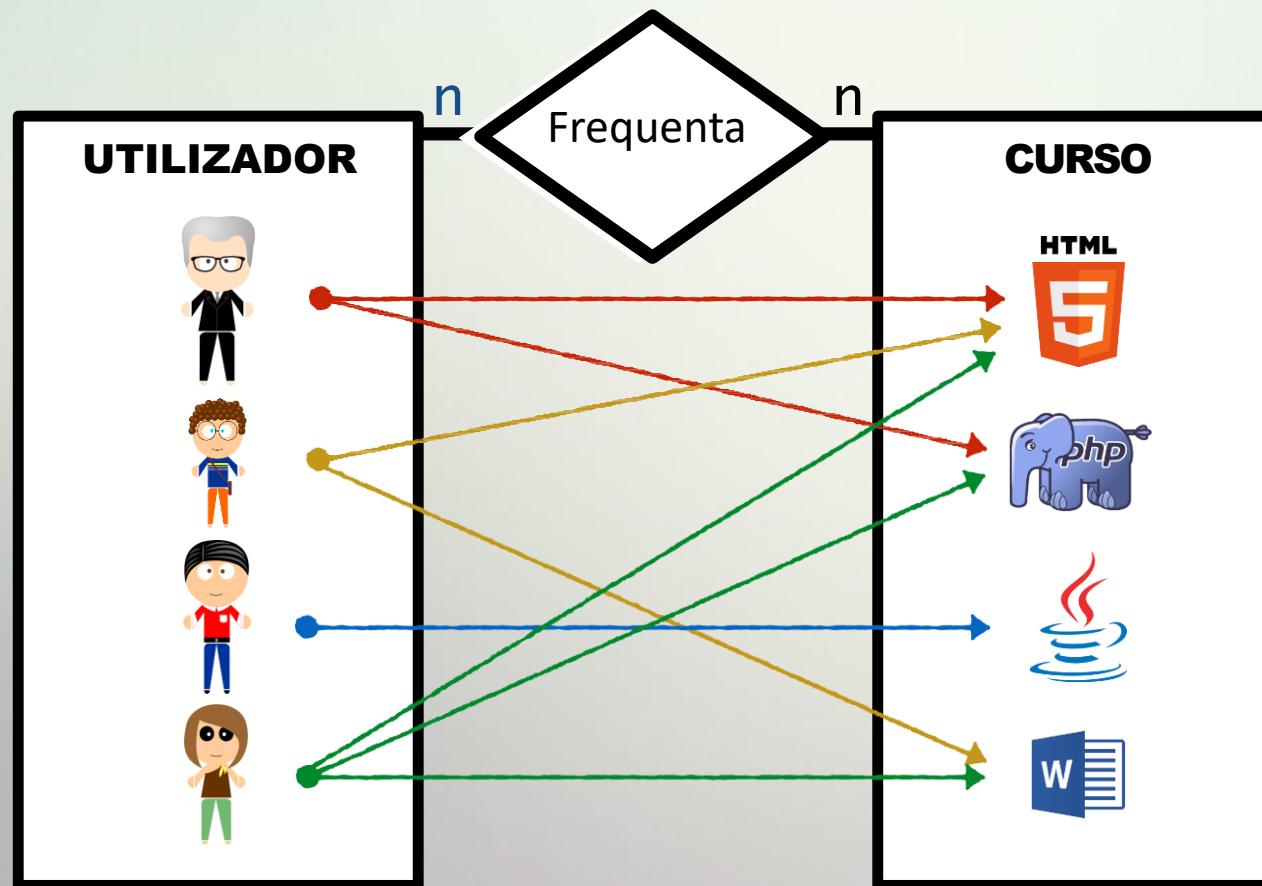
# DIAGRAMA E-R ou DER



# Frequenta



# Cardinalidade



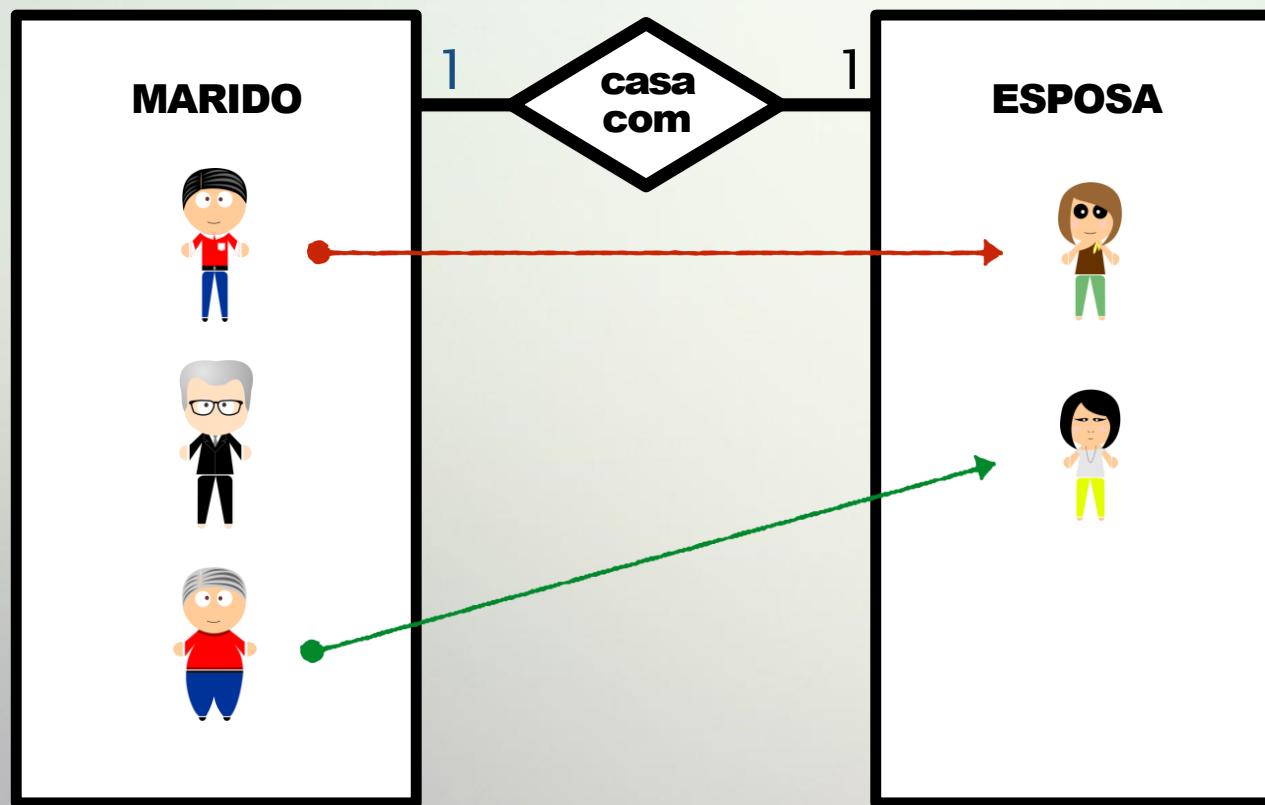
# Cardinalidade



 *Cardinalidade*

**muitos-para-muitos**

# Cardinalidade



# Cardinalidade

MARIDO

1

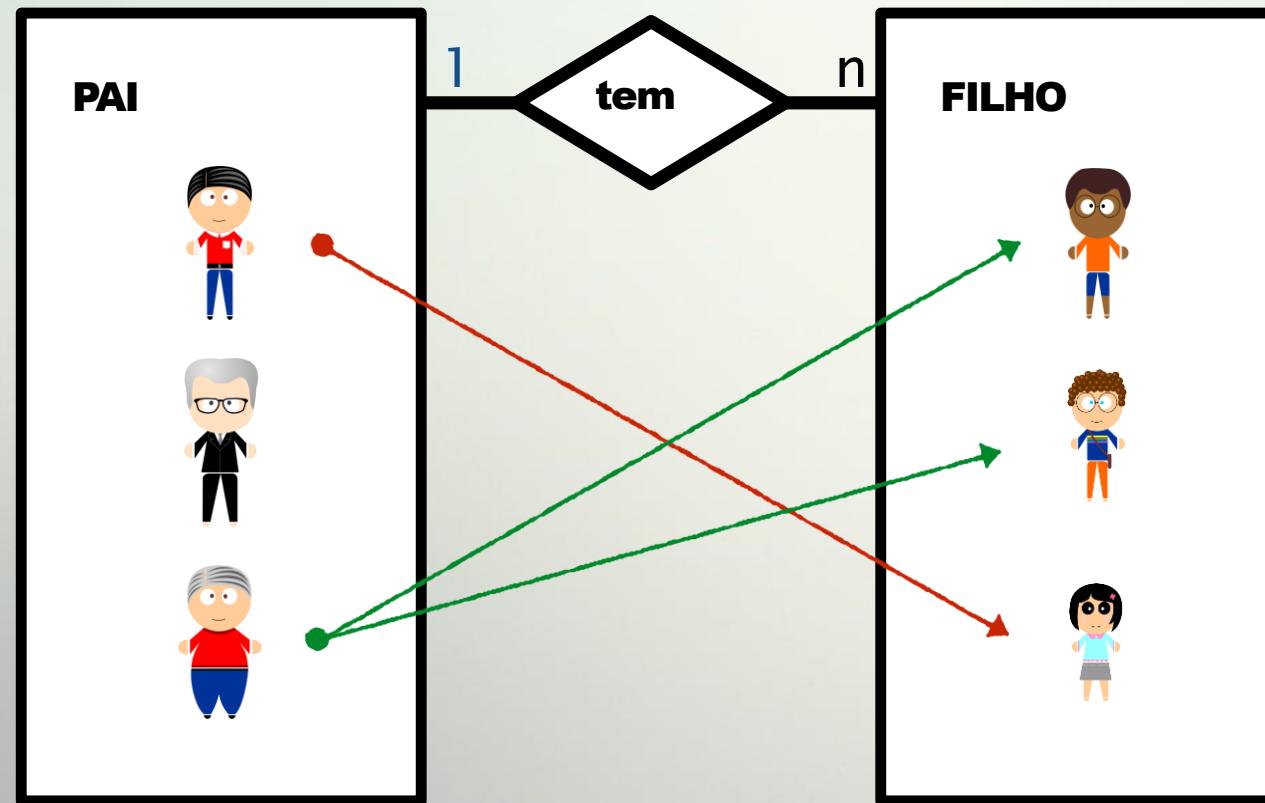
casa com

1

ESPOSA

um-para-um

# Cardinalidade

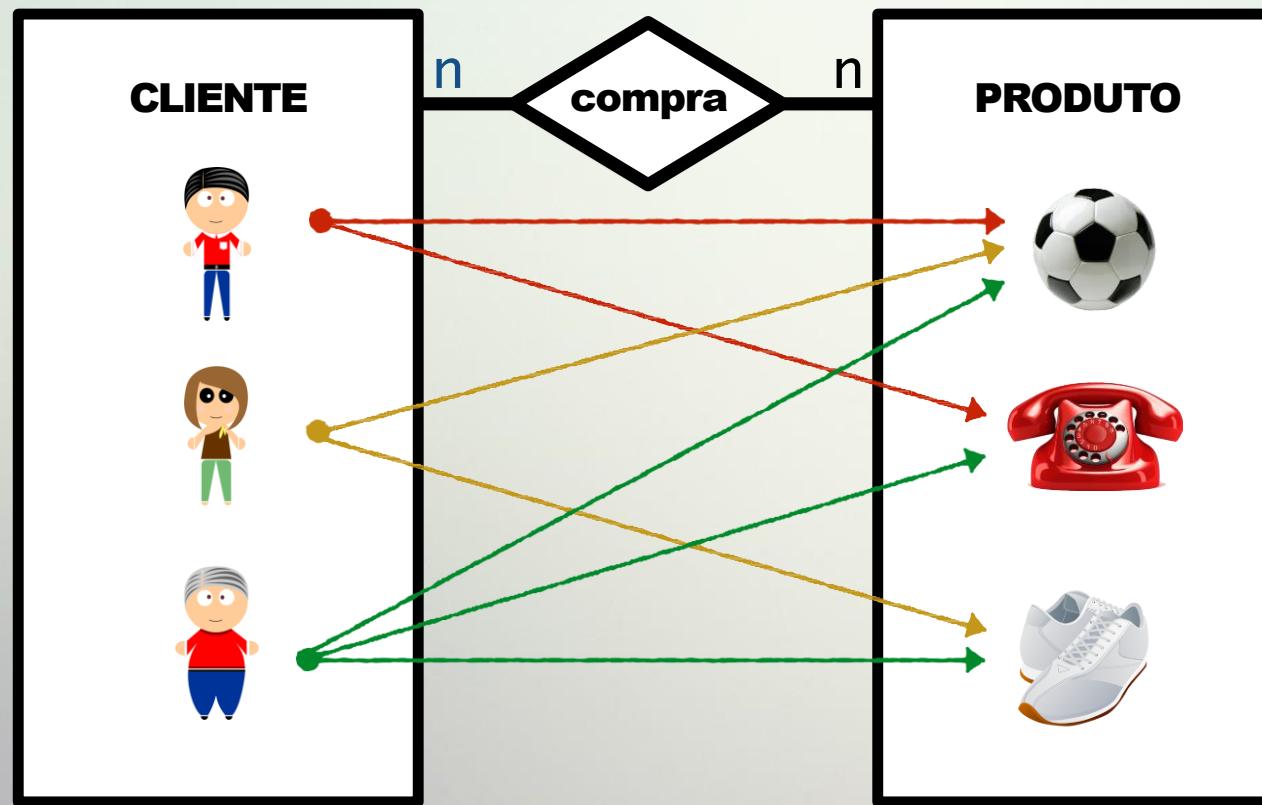


# Cardinalidade



um-para-muitos

# Cardinalidade

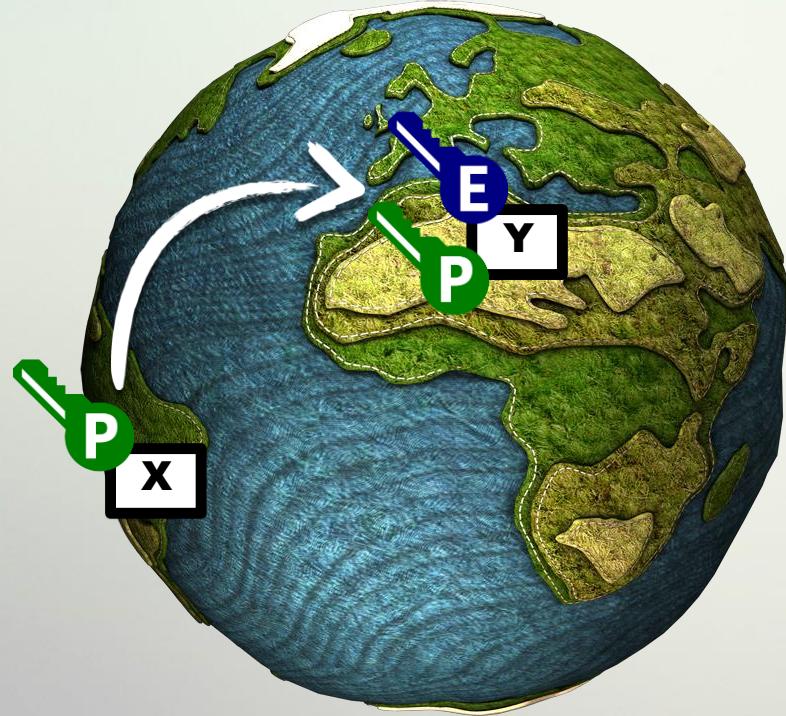


# Cardinalidade

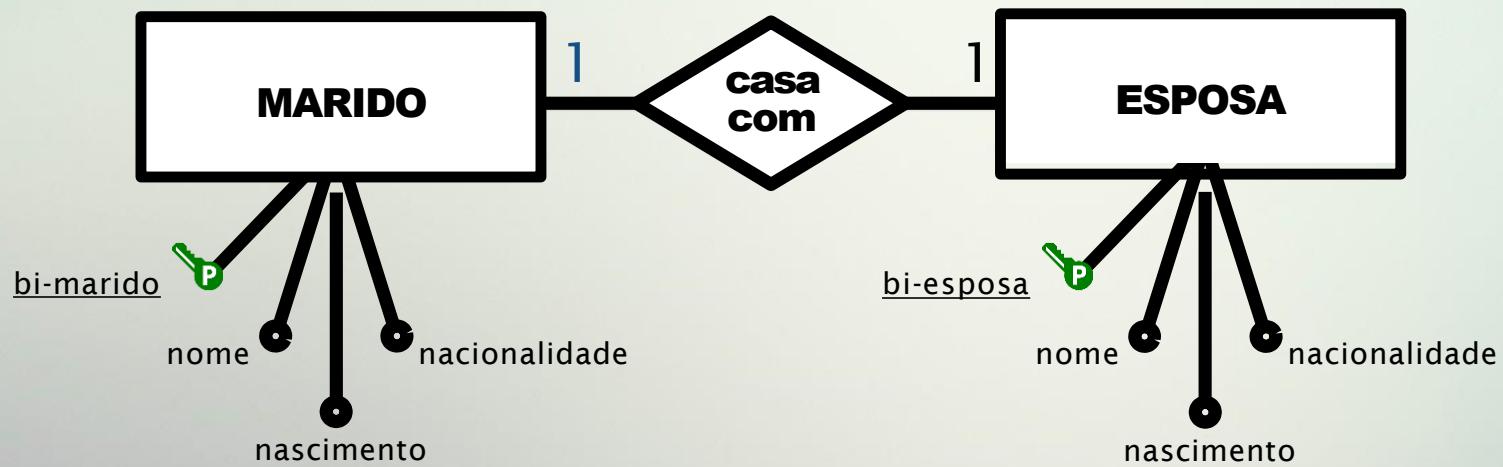


muitos-para-muitos

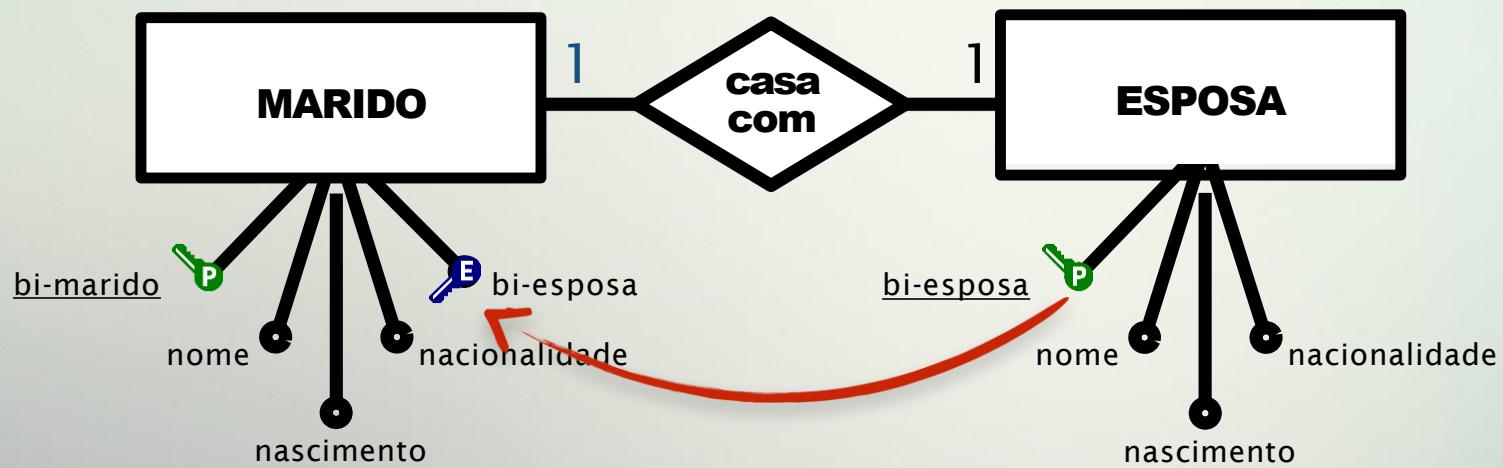
# Chave estrangeira



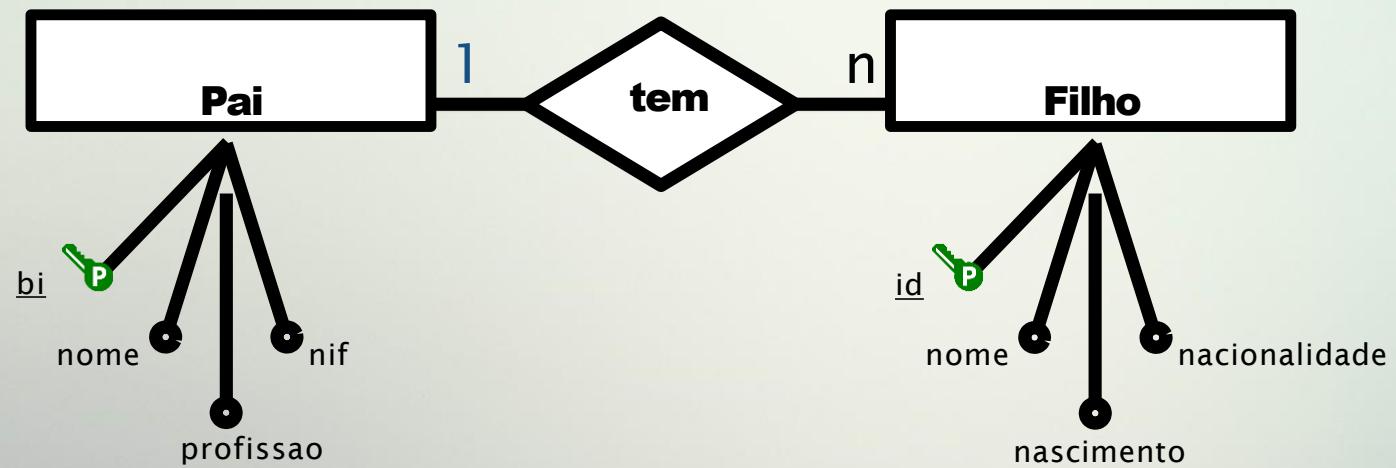
# Relação 1-1



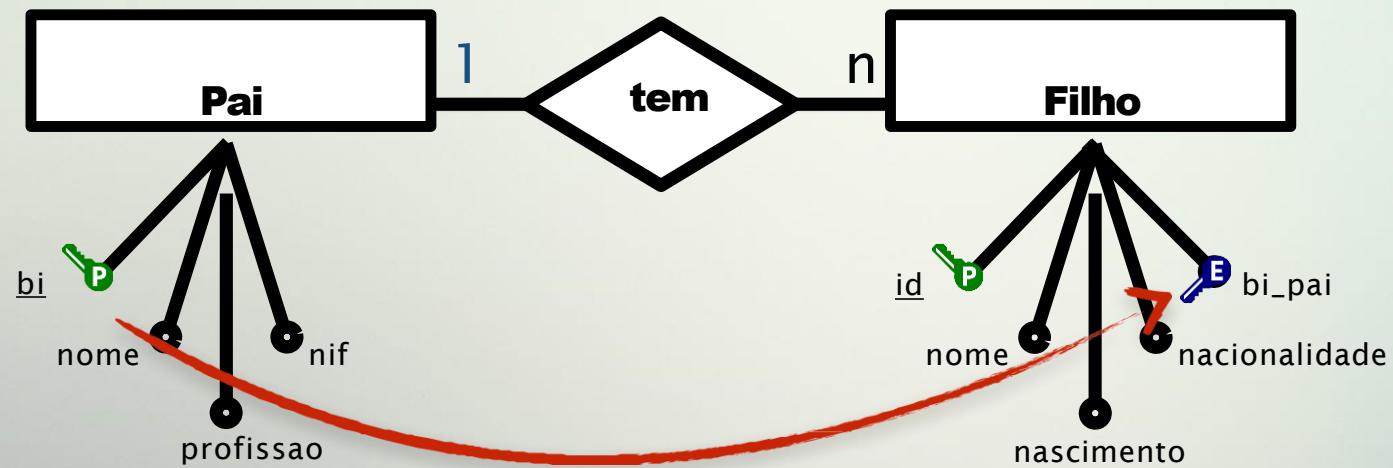
# Relação 1-1



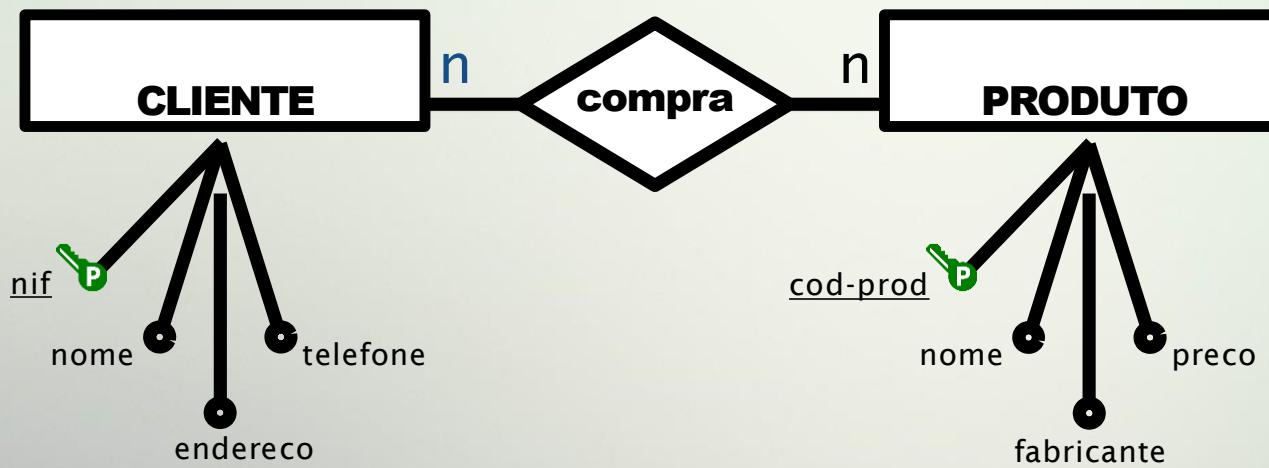
# Relação 1-N



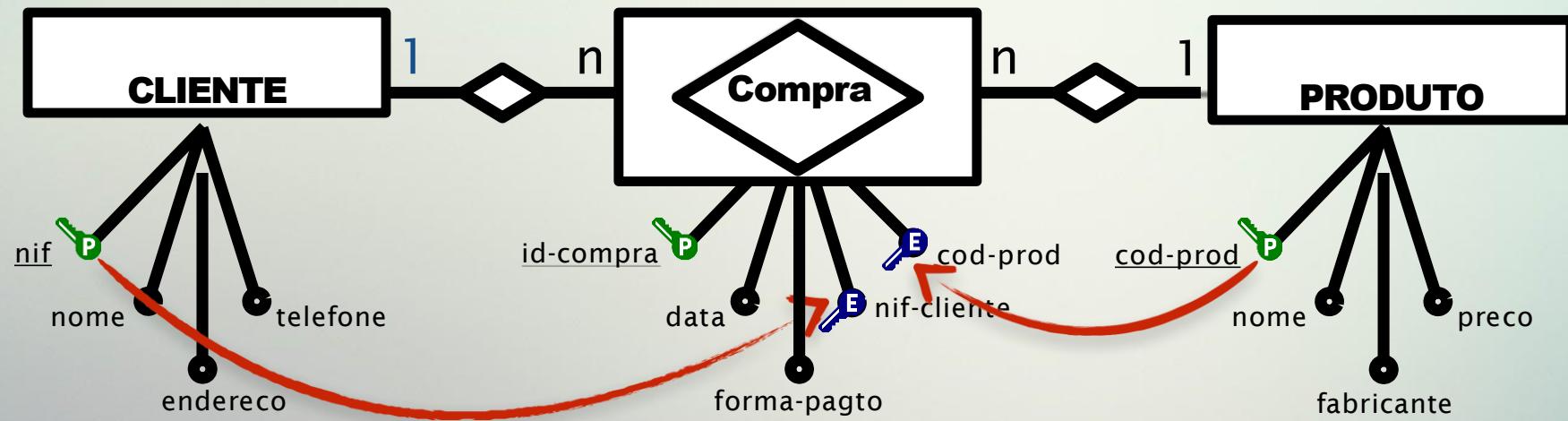
# Relação 1-N



# Relação N-N



# Relação N-N





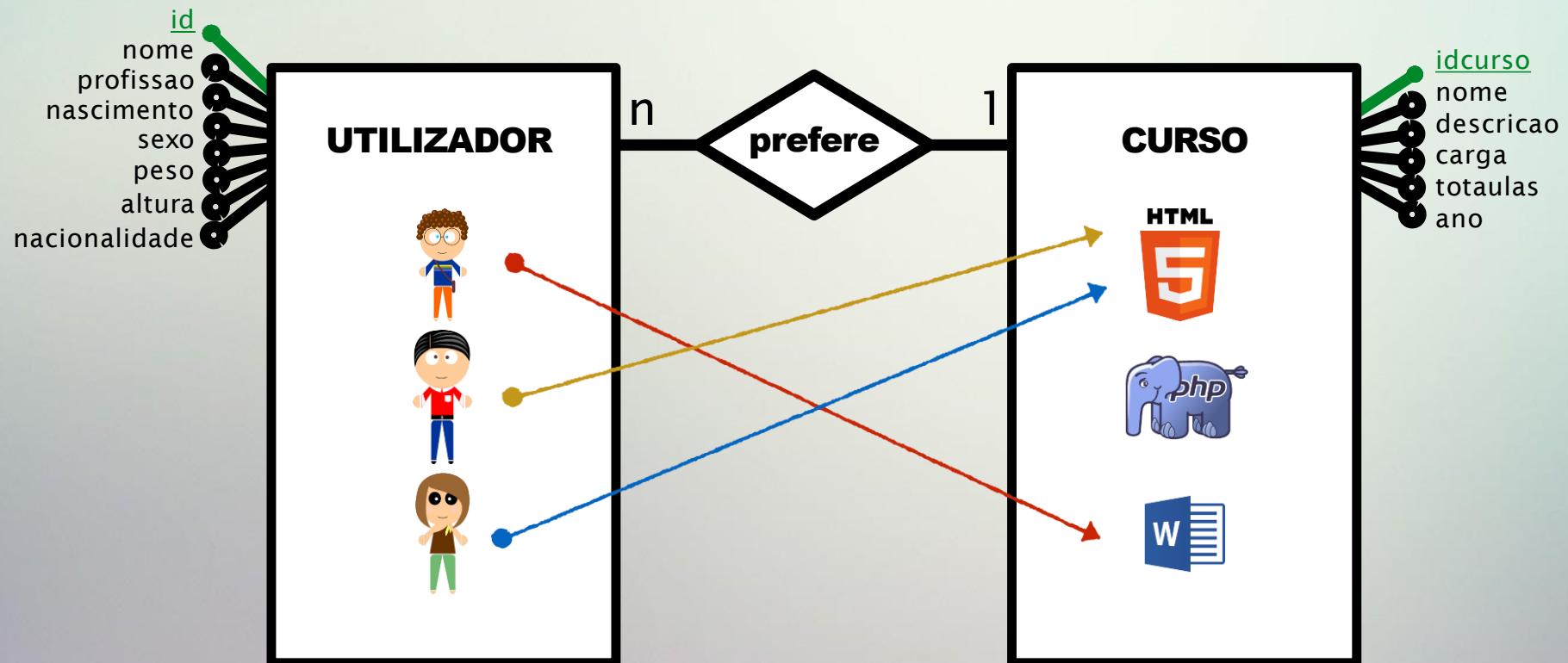
**MyISAM  
InnoDB  
XtraDB**

# **ACID**

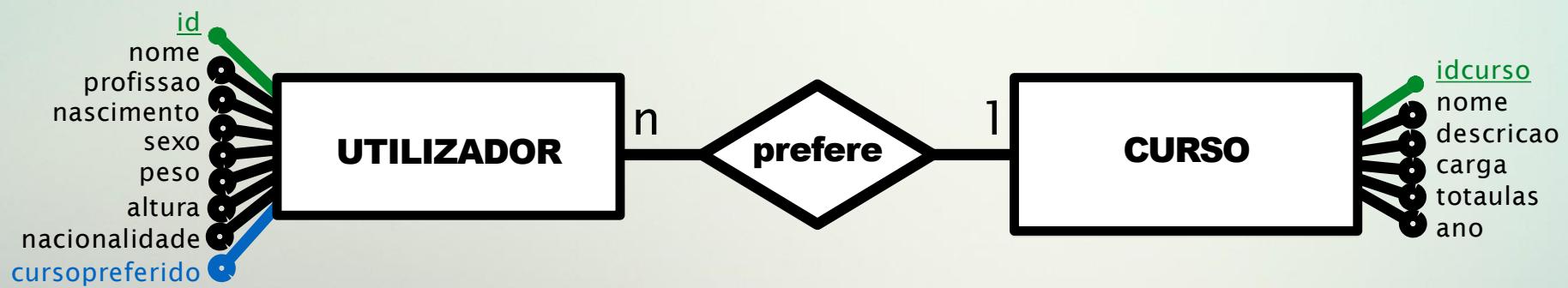
Atomicidade  
Consistência

Durabilidade  
Isolamento

# Exemplo – Como relacionar?



# Relação 1-N



Adicionar a chave primária da tabela curso à tabela utilizador

# Adicionando a Foreign Key

```
ALTER TABLE utilizadores  
ADD COLUMN cursopreferido int;
```

```
ALTER TABLE utilizadores  
ADD FOREIGN KEY (cursopreferido)  
REFERENCES cursos(idcurso);
```

# Atualizar os Registros

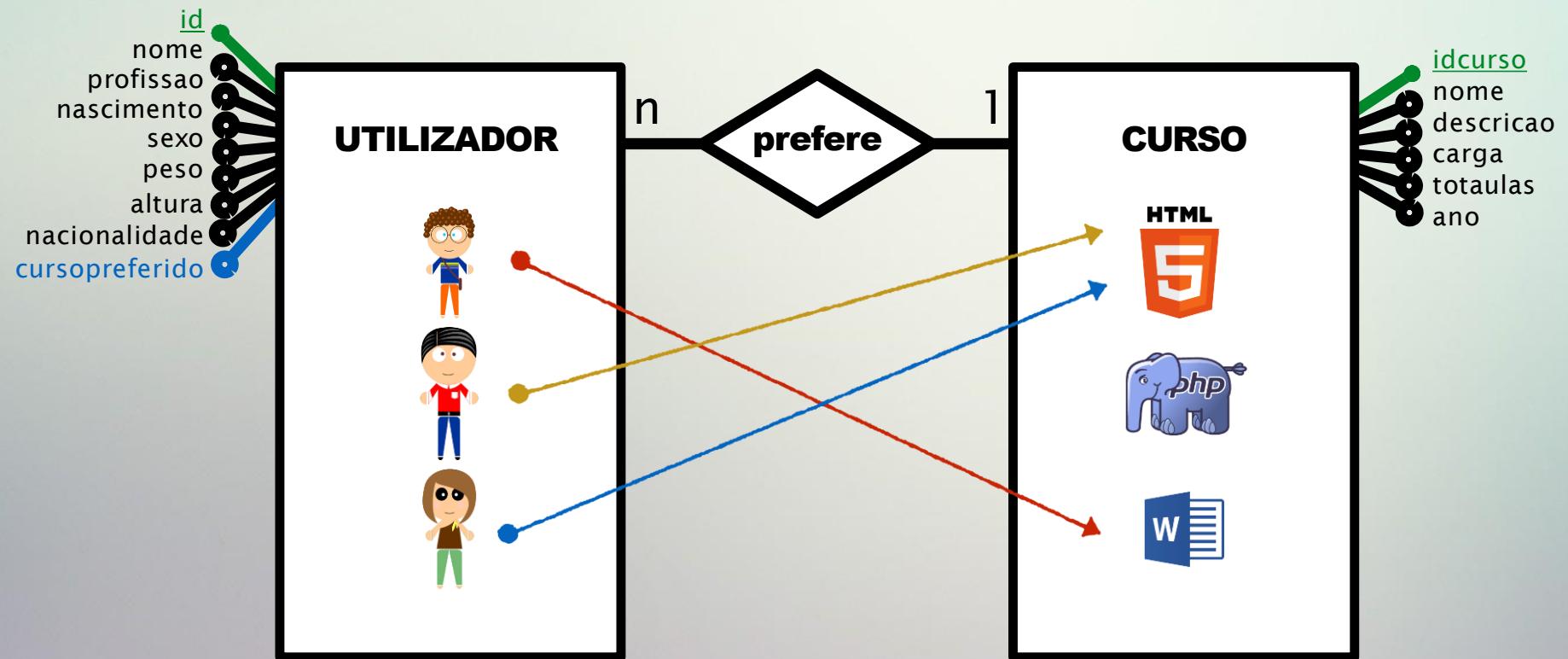
```
UPDATE utilizador  
SET cursopreferido = '5'  
WHERE id = '1';
```



# Integridade Referencial

```
DELETE FROM cursos  
WHERE idcurso = '6';
```

# Integridade Referencial



# Junções

```
SELECT nome, cursopreferido  
FROM utilizadores;
```

```
SELECT nome, ano  
FROM cursos;
```

Mas como se vê o nome do utilizador e do curso na mesma consulta?

# Junções

```
SELECT utilizadores.nome, utilizadores.cursopreferido,  
cursos.nome, cursos.ano  
FROM utilizadores JOIN cursos;
```

```
SELECT utilizadores.nome, utilizadores.cursopreferido,  
cursos.nome, cursos.ano  
FROM utilizadores JOIN cursos  
ON utilizadores.cursopreferido = cursos.idcurso;
```

# Junções

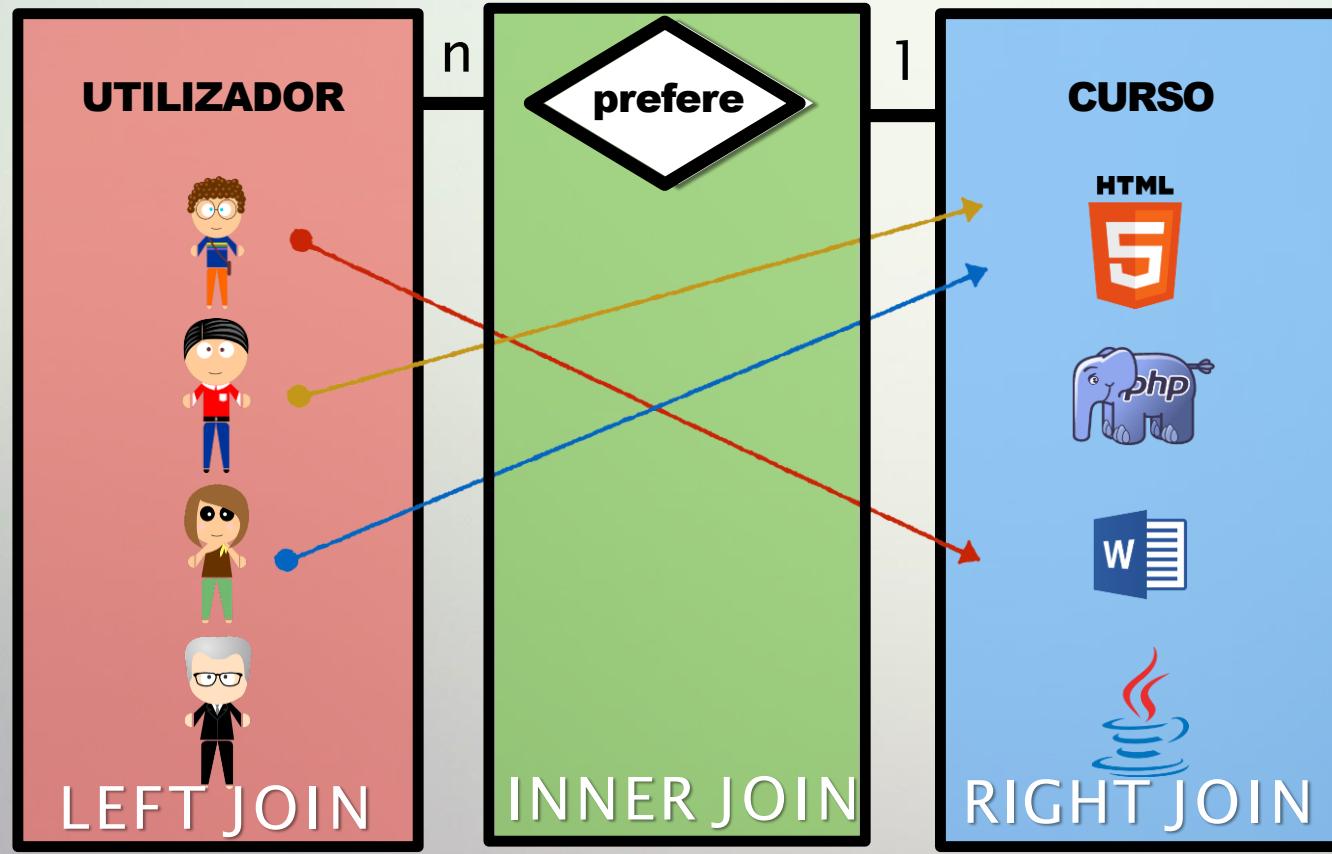
```
SELECT *
FROM utilizadores JOIN cursos
ON utilizadores.cursopreferido = cursos.idcurso;
```

```
SELECT utilizadores.*, cursos.nome
FROM utilizadores INNER JOIN cursos
ON utilizadores.cursopreferido = cursos.idcurso
ORDER BY utilizadores.nome;
```

# Junções

```
SELECT u.nome, u.cursopreferido, c.nome, c.ano  
FROM utilizadores as u INNER JOIN cursos as c  
ON u.cursopreferido = c.idcurso  
ORDER BY u.nome;
```

# Junções



# Junções

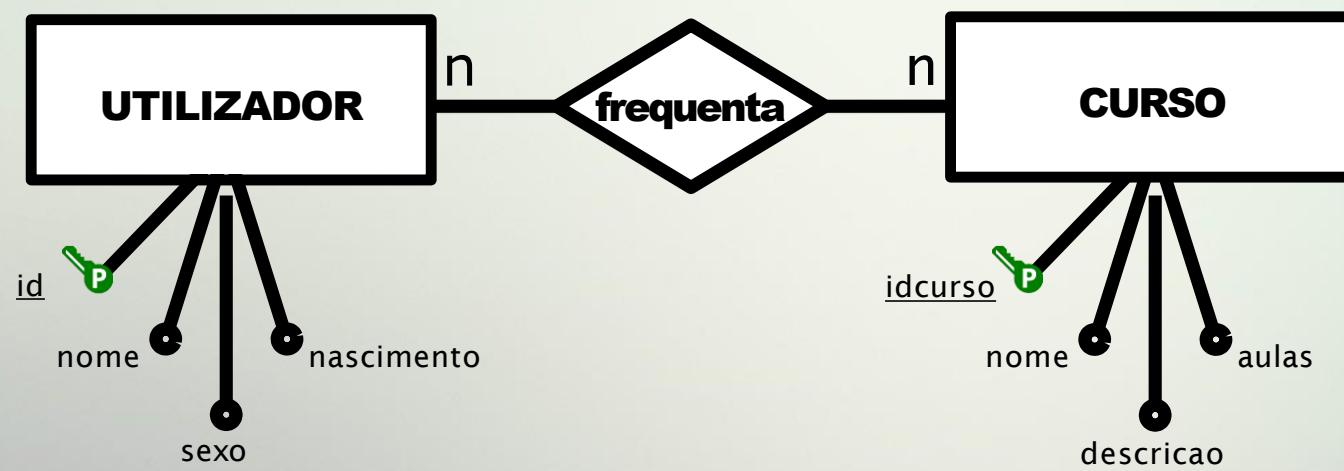
```
SELECT u.nome, u.cursopreferido, c.nome, c.ano  
FROM utilizadores as u LEFT OUTER JOIN cursos as c  
ON u.cursopreferido = c.idcurso;
```

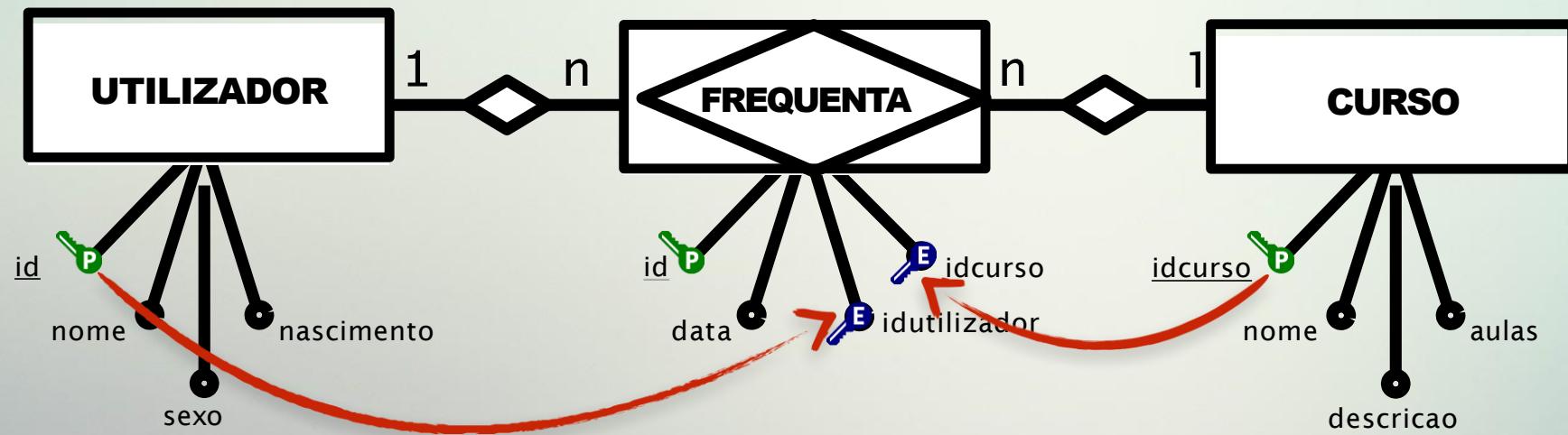
```
SELECT u.nome, u.cursopreferido, c.nome, c.ano  
FROM utilizadores as u LEFT JOIN cursos as c  
ON u.cursopreferido = c.idcurso  
ORDER BY u.nome;
```

# Junções

```
SELECT u.nome, u.cursopreferido, c.nome, c.ano  
FROM utilizadores as u RIGHT OUTER JOIN cursos as c  
ON u.cursopreferido = c.idcurso;
```

```
SELECT u.nome, u.cursopreferido, c.nome, c.ano  
FROM utilizadores as u RIGHT JOIN cursos as c  
ON u.cursopreferido = c.idcurso  
ORDER BY u.nome;
```

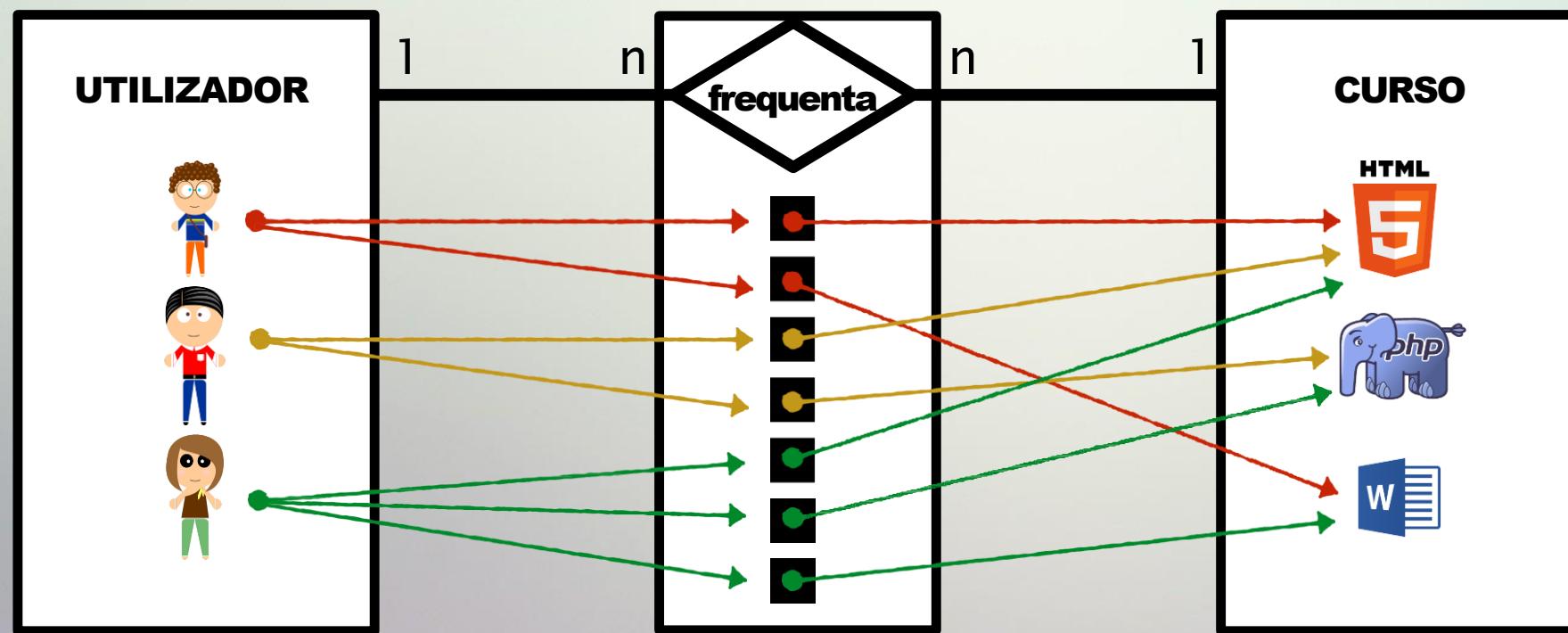




## Criando a Tabela extra

```
CREATE TABLE usuario_frequenta_curso (
    id int NOT NULL AUTO_INCREMENT,
    data date,
    idusuario int,
    idcurso int,
    PRIMARY KEY (id),
    FOREIGN KEY (idusuario) REFERENCES usuarios(id),
    FOREIGN KEY (idcurso) REFERENCES cursos(idcurso)
) DEFAULT CHARSET = utf8mb4;
```

# Explicação da Relação



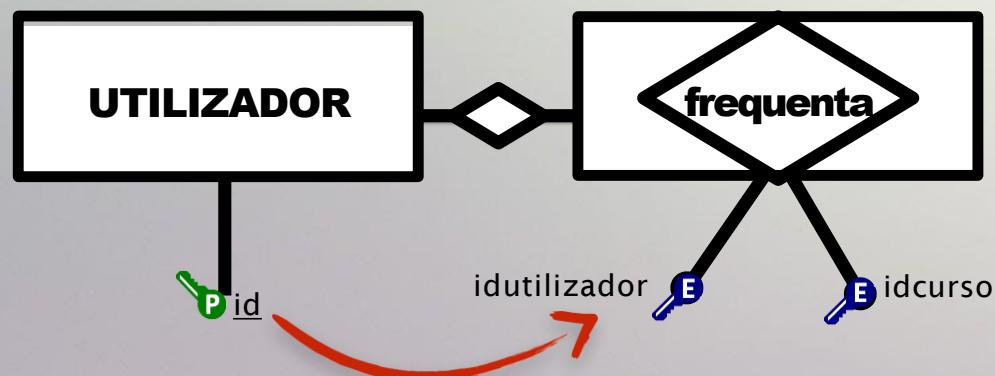


## Inserir os Registros

```
INSERT INTO utilizadores_frequenta_curso  
(id, data, idutilizador, idcurso)  
VALUES  
(DEFAULT, '2014-03-01', '1', '2'),
```

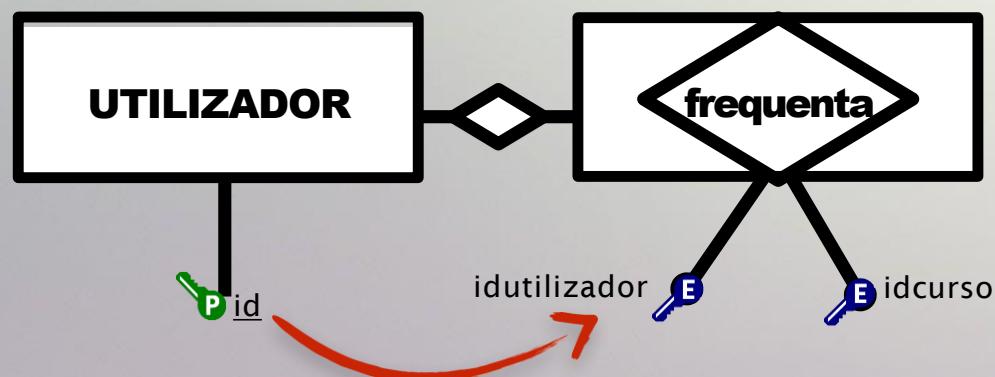
# Junções

```
SELECT *
FROM utilizadores u JOIN utilizadores_frequenta_curso ufc
ON u.id = ufc.udutilizador
```



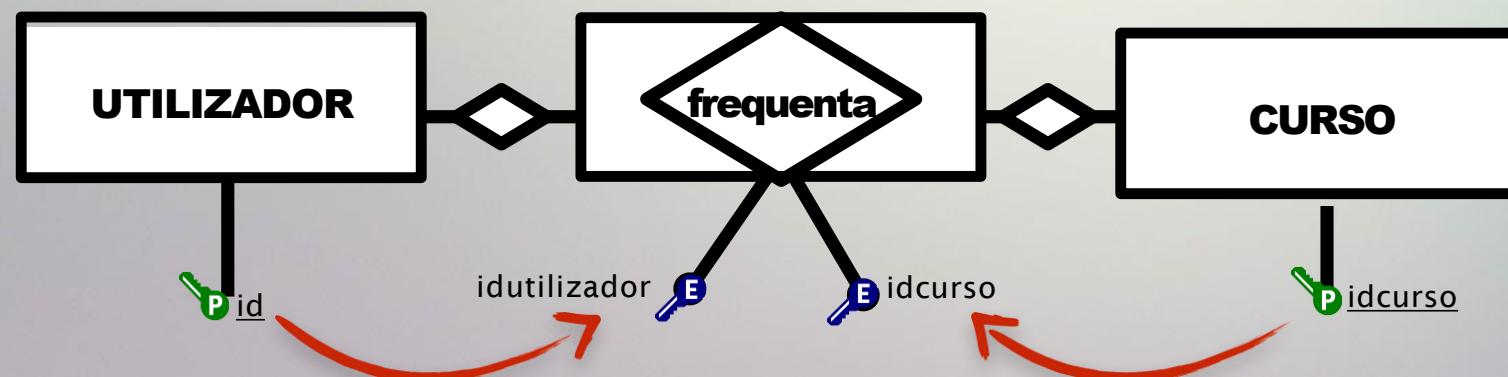
# Junções

```
SELECT u.id, u.nome, ufc.idutilizador, idcurso  
FROM utilizadores u JOIN utilizadores_frequenta_curso ufc  
ON u.id = ufc.udutilizador
```



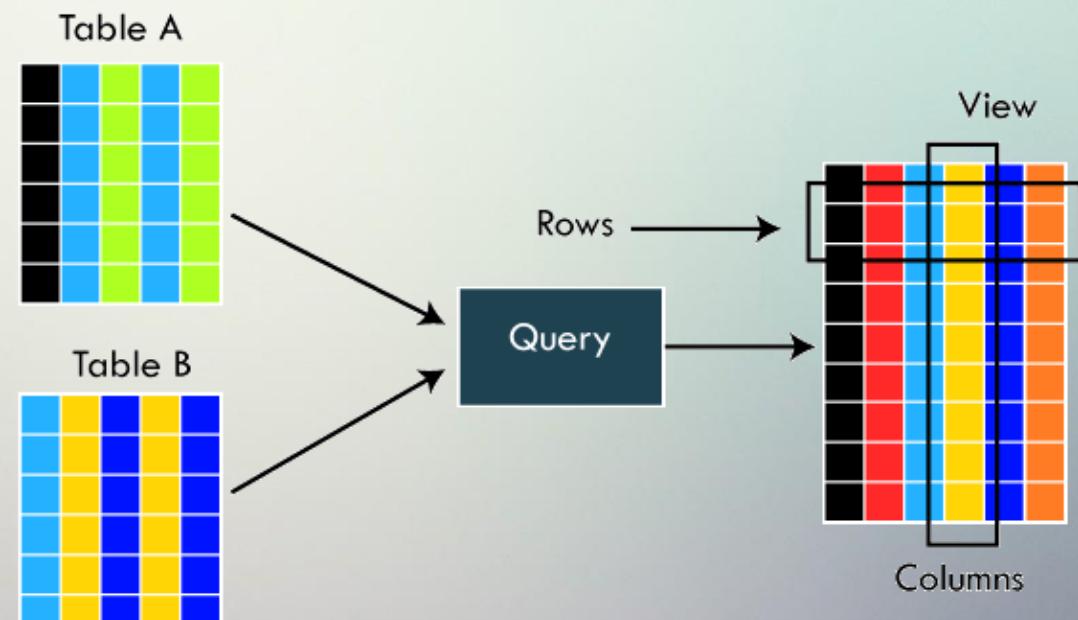
# Junções

```
SELECT u.id, u.nome, ufc.idutilizador, idcurso  
FROM utilizadores u  
JOIN utilizadores_frequenta_curso ufc  
ON u.id = ufc.idutilizador  
JOIN cursos c  
ON ufc.idcurso = c.idcurso;
```



# VIEW

- Objeto que pertence a um banco de dados.
- Definida com base em declarações SELECT.
- Retorna uma visualização de dados de uma ou mais tabelas.
- Conhecida como "tabela virtual".
- Podem ser atualizáveis em alguns casos.
- Suportam operações INSERT, UPDATE e DELETE.
- Alterações refletem diretamente nas tabelas base.
- Restringe dados para maior segurança.
- Simplifica consultas complexas.



# VIEW

```
CREATE VIEW nome_da_view  
AS expressao_select;
```

```
ALTER VIEW nome_da_view  
AS expressao_select;
```

```
DROP VIEW nome_da_view;
```

```
RENAME TABLE nome_view  
TO novo_nome_view;
```

## EXEMPLO

```
CREATE VIEW nome_da_view  
AS  
SELECT * FROM nome_tabela;
```



# EXERCÍCIO VIEW



- 1) Crie uma view `utilizadores_seu_nome` que mostre os nomes e as nacionalidades
- 2) Insira um novo registo na tabela `utilizadores`
- 3) Liste os dados da view `utilizadores_seu_nome`
- 4) Insira um novo registo na view
- 5) Liste os dados da tabela `utilizadores_seu_nome`

A que conclusão chegou?



# GESTÃO UTILIZADORES



```
CREATE USER nome_utilizador@hostname  
IDENTIFIED BY 'password';
```

```
CREATE USER nome_utilizador;
```

```
ALTER USER nome_utilizador@localhost  
IDENTIFIED BY 'nova_password';
```

# GESTÃO UTILIZADORES

```
SET PASSWORD FOR 'user'@'host' = PASSWORD('nova password');

DROP USER nome_utilizador@localhost;

RENAME USER nomeUtilizador TO novoNomeUtilizador;
```

## Outras funções

```
COMMENT 'informações sobre o utilizador';
PASSWORD EXPIRE
FAILED_LOGIN_ATTEMPTS 4 PASSWORD_LOCK_TIME 2;

SELECT user, host
FROM mysql.user; #consultar os utilizadores da base de dados
```

# EXERCÍCIOS



Crie um utilizador com o seu nome para o acesso de qualquer local com a password 123.

Depois do utilizador criado, defina uma nova password.

No Workbench crie uma nova instância de ligação com o utilizador que acabou de criar e faça a ligação ao servidor de base de dados.

Nota alguma diferença do seu utilizador para o utilizador root?



# Base de Dados – Objetivo?

- A base de dados tem como objetivo o armazenamento de diversos tipos de informação relacionada, de tal forma que a sua atualização e consulta possa ser efetuada num curto espaço de tempo.

# Base de Dados - Objetivo

- Qual será a base de dados mais vantajosa, em forma de papel ou a informatizada? Porquê?



# Base de Dados - Conceito

- Estrutura que permite armazenar informação sob a forma de um ou vários ficheiros de dados.
- Esta informação deverá estar relacionada entre si.
- O planeamento de uma base de dados permite que o armazenamento de dados seja feito de forma eficiente.

# Sistemas de Gestão de Base de Dados (SGBD) ou Data Management Software (DMS)

- Programa ou conjunto de programas que possibilitam a criação e manipulação de bases de dados (inserção, eliminação, alteração e consulta dos dados).
- Os dados são independentes dos programas que os manipulam.
- O **Access** e o **Workbench** são SGBD.

# Base de Dados

- São tipicamente representadas por **tabelas** (*entidade*) – onde são guardados os dados.
- As tabelas são compostas por:
  - Colunas que definem os **campos** (atributos da entidade)
  - Linhas que definem os **registos**
  - Nas células estão os **dados**

# Exemplo de uma tabela

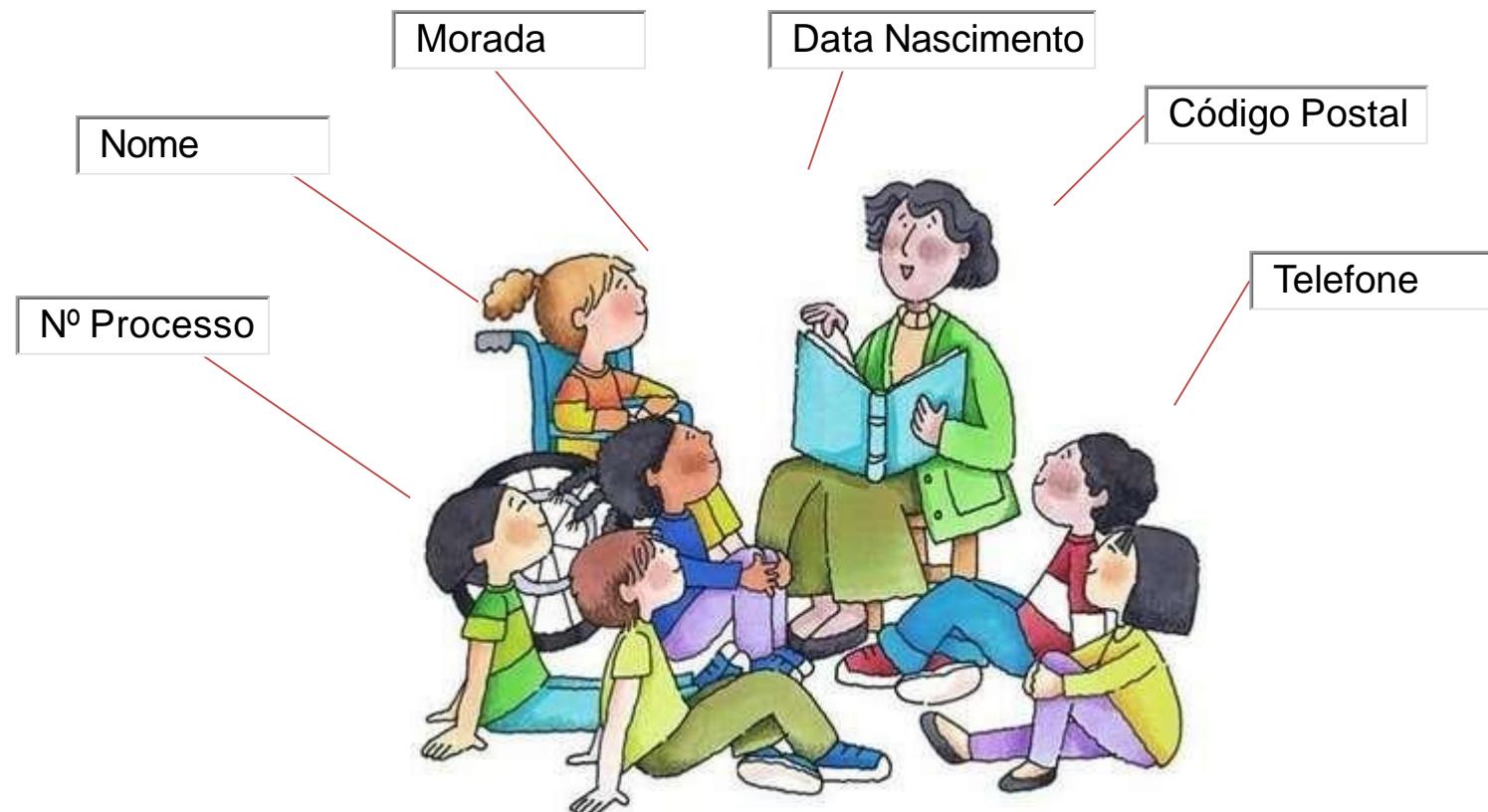
Campos

Registros

Dado

Código	Descrição	Preço	Unidade	IVA	Desconto	Última Data
1	Parque	25,00 €	M2	25	10	20-11-2012
2	Taco	20,00 €	M2	25	0	20-11-2012
3	Soalho	15,00 €	M2	25	20	30-11-2012
4	Cola	3,00 €	L	25	5	28-11-2012
5	Pregos	7,50 €	KG	25	0	29-11-2012
6	Cera	4,00 €	KG	25	0	29-11-2012
7	Verniz	5,50 €	L	25	5	29-11-2012
8	Mão de Obra	1,00 €	H	25	0	30-11-2012

# Criar campos para uma tabela de Alunos



# Problemas frequentes na elaboração de uma BD

- Como representar uma situação do mundo real através de conceitos teóricos?
  - Por **exemplo**, como representar:
    - Um fornecedor (xpto, Ld<sup>a</sup>)
    - Um produto (computador)
    - Uma encomenda (um computador é encomendado ao fornecedor xpto, Ld<sup>a</sup> - relacionamento)

# Noção de Relação

- Uma relação consiste na associação estabelecida entre campos comuns de duas tabelas, garantindo assim a consistência da informação.

**Entidade:** Clientes

**Campos:**

Código  
Nome  
Morada  
Cód Postal  
Telefone  
Email

**Entidade:** Cód. Postais

**Campos:**

Código  
Localidade

Relação

# Noção de Chave

- A **chave** de uma tabela é constituída por um ou mais campos que possam ser utilizados como identificadores de cada um dos registos. Ou seja que permitam identificar um registo de forma unívoca (única - um e apenas um registo).

# Tipos de Atributos Chave:

- Atributos Identificadores (**chave primária**)
  - Identifica inequivocamente um registo numa entidade
  - Exemplo: BI na entidade pessoa, matricula na entidade carro
- Atributos candidatos (**chave candidata**)
  - Atributos que estão em condições de ser chaveprimária
  - Exemplo: Nº contribuinte na entidade pessoa
- Chave Estrangeira ou Externa
  - Atributo ou conjunto de atributos que aparecem como chaveprimária numa outra tabela (normalmente campos do relacionamento entre tabelas)

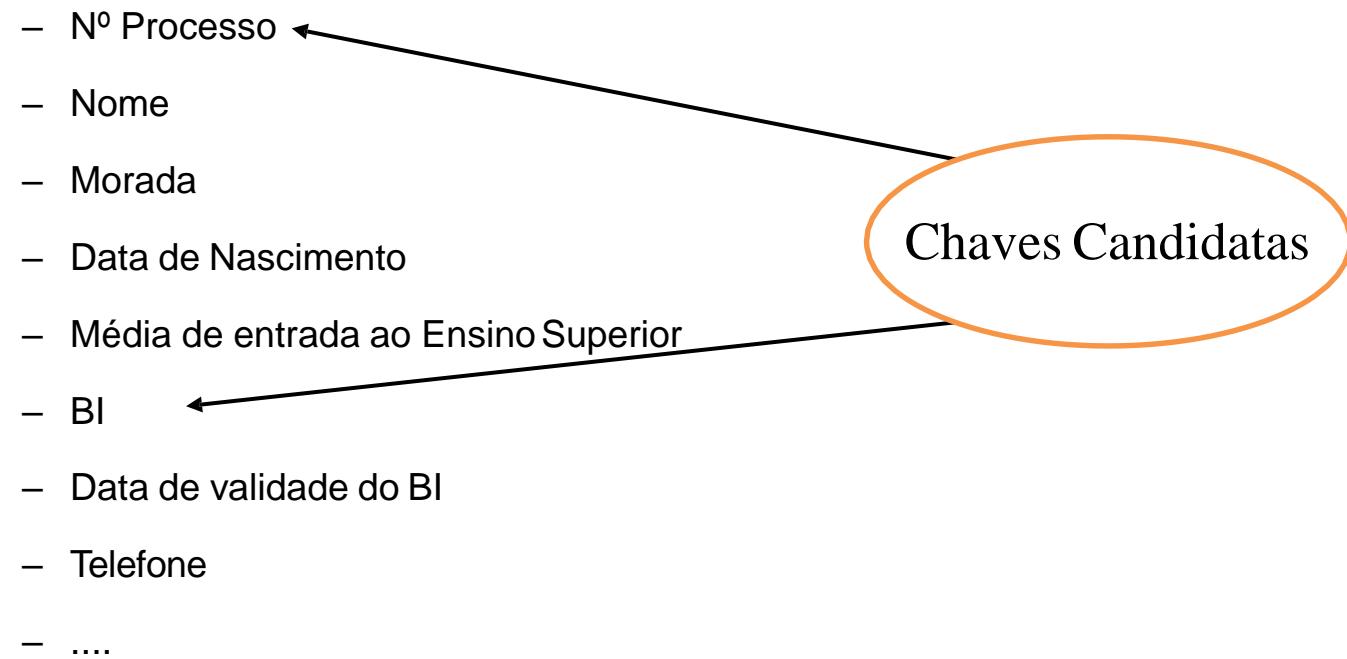
# As chaves podem ser:

- Simples - constituídas apenas por um atributo
  - Exemplo: Nº aluno - na entidade aluno
- Compostas - constituídas por mais de um atributo
  - Exemplo: Na entidade Avaliação pode ser definida a seguinte chave  
→ Nº aluno + Código da Disciplina + Época + data

# Exemplo de Atributos:

- Na entidade aluno podem definir-se os atributos:

- Nº Processo
- Nome
- Morada
- Data de Nascimento
- Média de entrada ao Ensino Superior
- BI
- Data de validade do BI
- Telefone
- ....



Chaves Candidatas

# Nomenclatura para representar uma entidade

NOME\_DA\_ENTIDADE(atributo1, atributo2, atributo3, ...)

- Os atributos sublinhados indicam a chave primária
- Exemplo: T\_Aluno (Procaluno, nome, dataNasc,...)

# Deve-se evitar nas bases de dados

- **Redundância de informação:** repetição desnecessária de informação em vários locais

NumAluno	NomeAluno	Morada	Contacto	Disciplina	Epoca	Nota
1000	Ana Maria	Leiria	933552266	Informática I	Frequência	8
1000	Ana Maria	Leiria	933552266	Informática I	Exame	10
1000	Ana Maria	Leiria	933552266	Informática I	Recurso	15
1000	Ana Maria	Leiria	933552266	Economia	Frequência	16
1001	João Silva	Porto	225566772	Informática I	Frequência	13
1001	João Silva	Porto	225566772	Matemática	Exame	19

# Como dividir a informação para evitar a redundância?

NumAluno	Nome	Localidade	ContactoTelefonico	Disciplina	Data	Época	Nota
1	Ana Maria	Leiria	244 111 222	Informática I	12-02-2007	Frequência	12
1	Ana Maria	Leiria	244 111 222	Contabilidade	12-05-2006	Frequência	8
1	Ana Maria	Leiria	244 111 222	Contabilidade	10-02-2007	Exame	9
3	Tiago Alves	Lisboa	212 333 444	Informática I	12-02-2007	Frequência	10
4	Rui Oleiro	Aveiro	222 222 222	Informática I	12-02-2007	Frequência	19

Alunos

NumAluno	Nome	Localidade	ContactoTelefonico
1	Ana Maria	Leiria	244 111 222
2	José Gomes	Parceiros	244 222 333
3	Tiago Alves	Lisboa	212 333 444
4	Rui Oleiro	Aveiro	222 222 222

Disciplinas

Código	Nome
1	Informática I
2	Informática II
3	Matemática I
4	Contabilidade
5	Direito Constitucional

Notas

Nº do Aluno	Código da Disciplina	Data	Época	Nota
1	1	12-02-2007	Frequência	12
1	4	12-05-2006	Frequência	8
1	4	10-02-2007	Exame	9
3	1	12-02-2007	Frequência	10
4	1	12-02-2007	Frequência	19

# Objetivos a atingir numa BD

- Incluir toda a informação relevante;
- Evitar redundância de informação;
- Assegurar a consistência e a integridade da informação na utilização da BD;

# Exemplo Planificação de uma Base de Dados

- Os empregados de um gabinete de advogados decidiram criar uma pequena biblioteca para partilharem os livros que cada um tem.
- Numa primeira fase pretendem uma base de dados que registe quem (num\_funcionario, nome) disponibiliza os livros (oferecidos ou somente emprestados) e a identificação de cada livro. Considere que todos os livros têm códigos diferentes.