

# Modelo relacional

Fernando Lobo

Base de Dados, Universidade do Algarve

1 / 17

## Modelo relacional

- Modelo simples e intuitivo baseado no conceito matemático de relação.
- Inventado em 1970 por Edgar Codd.
- Utilizado pela maioria dos SGBDs.
- SQL é baseado no modelo relacional.

2 / 17

## Modelo relacional

- Uma relação é como se fosse uma tabela com linhas e colunas.
- A diferença é que:
  - ▶ uma relação não pode ter linhas repetidas.
- Muitas vezes irei usar o termo tabela para me referir a uma relação (mas devem estar conscientes de que tabela e relação são coisas distintas).
- BD = conjunto de relações/tabelas.

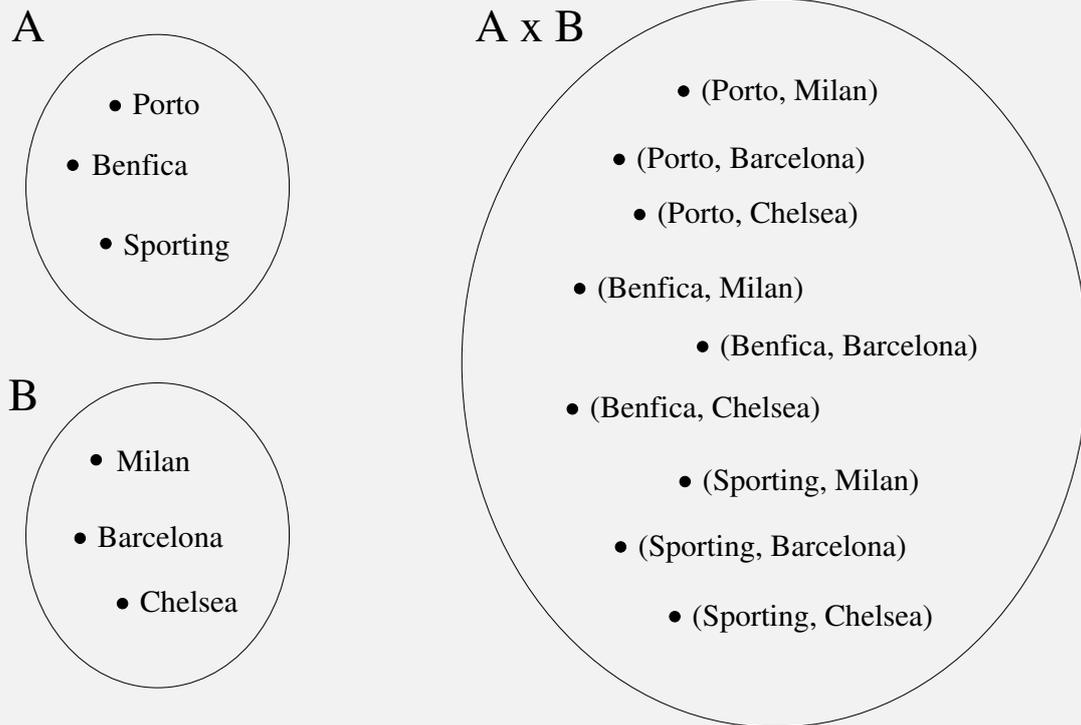
3 / 17

## Definição formal de relação

- Dados  $n$  conjuntos  $C_1, C_2, \dots, C_n$  (não necessariamente distintos),  $R$  é uma **relação** sobre estes  $n$  conjuntos, se for um conjunto de tuplos da forma  $(v_1, v_2, \dots, v_n)$  em que:
  - ▶  $v_1 \in C_1$
  - ▶  $v_2 \in C_2$
  - ▶ ...
  - ▶  $v_n \in C_n$
- Por outras palavras,  $R$  é subconjunto do Produto Cartesiano  $C_1 \times C_2 \times \dots \times C_n$ .

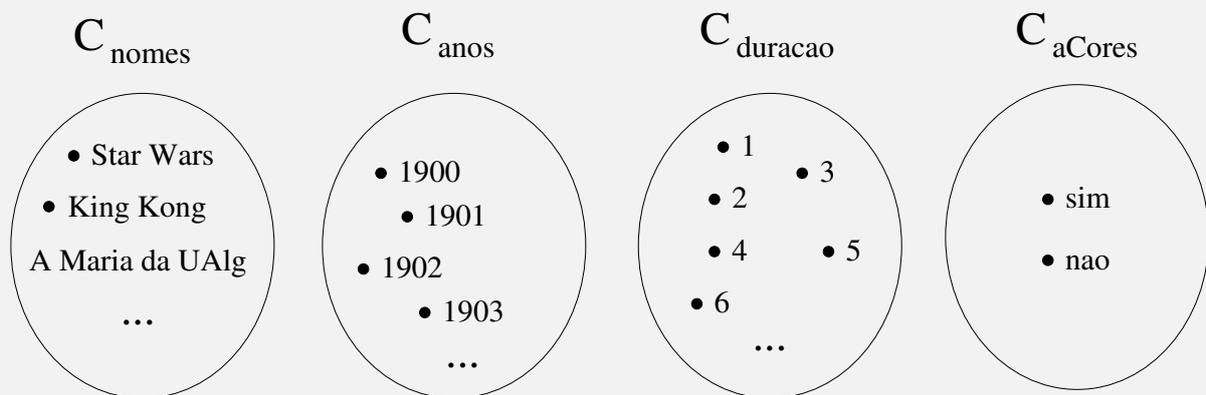
4 / 17

## Produto Cartesiano. Exemplo:



5 / 17

## Outro exemplo



- NOTA: Conjuntos podem ser infinitos.

6 / 17

## Outro exemplo (cont.)

$$C_{\text{nomes}} \times C_{\text{anos}} \times C_{\text{duracao}} \times C_{\text{aCores}}$$

- (Star Wars, 1900, 1, sim)
- (Star Wars, 1900, 1, nao)
- (Star Wars, 1900, 2, sim)
- (Star Wars, 1900, 2, nao)
- (Star Wars, 1900, 3, sim)

...

- Produto cartesiano dá o conjunto de todos os possíveis tuplos  $(v_1, v_2, v_3, v_4)$ , em que  $v_1 \in C_{\text{nomes}}$ ,  $v_2 \in C_{\text{anos}}$ ,  $v_3 \in C_{\text{duracao}}$ ,  $v_4 \in C_{\text{aCores}}$ .

7 / 17

## Uma possível relação sobre $C_{\text{nomes}}$ , $C_{\text{anos}}$ , $C_{\text{duracao}}$ , $C_{\text{aCores}}$

- (Star Wars, 1977, 124, sim)
- (King Kong, 1930, 120, nao)
- (Moulin Rouge, 2001, 124, sim)

8 / 17

## Dá jeito visualizar a relação como uma tabela

nome	ano	duração	aCores
Star Wars	1977	124	sim
King Kong	1930	120	não
Moulin Rouge	2001	124	sim

- Nome das colunas são atributos.
- tuplo = linha.

9 / 17

## Estas 2 tabelas representam a mesma relação. Porquê?

nome	ano	duração	aCores
Lion King	1997	108	sim
Star Wars	1977	124	sim
King Kong	1930	120	não
Lion King	1997	108	sim
Lion King	1997	108	sim

ano	nome	aCores	duração
1930	King Kong	não	120
1977	Star Wars	sim	124
1997	Lion King	sim	108

10 / 17

# Esquema da BD = descrição completa da estrutura da BD

- Nome das relações/tabelas
- Nome dos atributos
- Domínio dos atributos (integer, string, ...)
- Chaves
- ...

11 / 17

## Instância da BD

- O conteúdo (os tuplos) de todas as relações de uma BD constitui uma instância da BD.
- A instância da BD muda com frequência.
- O esquema da BD é feito uma vez (por pessoas como vocês) e não costuma mudar.

12 / 17

## Definição de tabelas em SQL

```
CREATE TABLE <nome> (  
    <lista de elementos>  
);
```

- cada elemento consiste num atributo e no respectivo tipo de dados.
- os tipos mais comuns são:
  - ▶ INT ou INTEGER
  - ▶ REAL ou FLOAT
  - ▶ CHAR(n), VARCHAR(n)
  - ▶ BOOLEAN
  - ▶ DATE
  - ▶ TIME

13 / 17

## Exemplo: Tabela de actores

```
CREATE TABLE Actores(  
    nome            VARCHAR(50),  
    morada          VARCHAR(70),  
    sexo            CHAR,  
    dataNascimento DATE  
);
```

14 / 17

## DATE e TIME

- o formato de DATE é 'yyyy-mm-dd'.
- o formato de TIME é 'hh:mm:ss'.
  - ▶ poderá ter ainda fracções de segundo.
- Exemplos:
  - ▶ DATE '2008-01-05'  
(5 de Janeiro de 2008)
  - ▶ TIME '15:32:04.5'  
(4 segundos e meio depois das 3:32 da tarde)

15 / 17

## Declaração da chave

- um atributo ou lista de atributos pode ser declarado como chave em SQL utilizando **PRIMARY KEY**.
- Exemplo:

```
CREATE TABLE Actores(  
    nome            VARCHAR(50) PRIMARY KEY,  
    morada          VARCHAR(70),  
    sexo            CHAR,  
    dataNascimento DATE  
);
```

16 / 17

## Declaração da chave (cont.)

- No caso da chave ser composta, temos de especificar um elemento à parte.
- Exemplo:

```
CREATE TABLE Filmes(  
    nome          VARCHAR(50),  
    ano           INTEGER,  
    duracao       INTEGER,  
    aCores        BOOLEAN,  
    PRIMARY KEY (nome,ano)  
);
```