

Structured Query Language (SQL)

- É uma espécie de implementação da álgebra relacional.
- A diferença é que utiliza sacos (bags) em vez de conjuntos (sets).
- Contem alguns operadores que não existem em álgebra relacional.
- SQL é usado em quase todos os SGBDs.

Porquê SQL?

- É uma linguagem de muito alto nível.
- Bem mais adequada para manipular relações do que linguagens como o C ou Java.
- O programador não tem de se preocupar com detalhes físicos.
- As queries em SQL são otimizadas.

BD de Filmes

Estudios(nome, morada)

Realizadores(nome, categoria)

Filmes(nome, ano, duracao, aCores, nomeEstudio, nomeRealizador)

Actores(nome, morada, sexo, dataNascimento)

Participa(nomeFilme, anoFilme, nomeActor)

Queries simples (SELECT-FROM-WHERE)

SELECT atributos
FROM uma ou mais relações
WHERE condição

Quais são os filmes feitos em 1977?

```
SELECT nome  
FROM Filmes  
WHERE ano=1977;
```

nome

Star Wars

Saturday Night Fever

...

Operação em termos de álgebra relacional (cont.)

1. Começa com a relação que aparece a seguir a FROM.
2. Aplica o operador σ usando a condição do WHERE.
3. Aplica o operador π usando os atributos de SELECT.

Álgebra relacional vs. SQL

Álgebra Relacional:

$$\pi_{nome} \left(\sigma_{ano = 1977} (\text{Filmes}) \right)$$

SQL:

```
SELECT nome
FROM Filmes
WHERE ano=1977;
```

Outra maneira de pensar

Imaginem uma variável que percorre todos os tuplos da relação.

- Verifica se satisfaz a condição `WHERE`.
- Se sim, envia os atributos especificados em `SELECT` para o output.

Asterisco dá todos os atributos

```
SELECT *  
FROM Participa  
WHERE nomeFilme='Moulin Rouge';
```

nomeFilme	anoFime	nomeActor
Moulin Rouge	2001	Nicole Kidman
Moulin Rouge	2001	Ewan McGregor
Moulin Rouge	2001	Jim Broadbent

Mudar o nome a colunas

- utilizar **AS** '<novo nome>'

```
SELECT nome AS 'nome do filme'  
FROM Filmes  
WHERE ano=1977;
```

nome do filme

Star Wars

Saturday Night Fever

...

Operadores lógicos

- a cláusula **WHERE** pode ter operadores lógicos (AND, OR, NOT)
- pode-se usar os operadores relacionais habituais (=, <>, <, >, <=, >=)

```
SELECT nome, duracao  
FROM Filmes  
WHERE estudio='Disney' AND ano=1977;
```

Expressões em SELECTs

- pode-se usar expressões como elementos da cláusula SELECT

```
SELECT nome, ano,  
       duracao/60 AS 'duração em horas'  
FROM Filmes;
```

nome	ano	duração em horas
Star Wars	1977	2.07
Empire Strikes Back	1980	2.38
Return of the Jedi	1983	2.75
Moulin Rouge	2001	2.07

Padrões

- a condição WHERE pode especificar uma comparação entre uma string e um padrão.
- <atributo> LIKE <padrão>
<atributo> NOT LIKE <padrão>
- um padrão é uma string em que % significa “0 ou mais caracteres” e _ significa “1 caracter”

Exemplo

Quais são os filmes cujo nome contem a palavra King?

```
SELECT nome, ano
FROM Filmes
WHERE nome LIKE '%King%';
```

nome	ano
King Kong	1933
King Kong	1976
Lion King	1994

Sacos vs. conjuntos vs. listas

- um conjunto não tem elementos repetidos.
- mas um saco pode ter elementos repetidos.
- nos sacos e conjuntos, a ordem dos elementos não interessa.
- mas numa lista ou array já interessa.

Sacos vs. conjuntos vs. listas (cont.)

- $\{b, a, a\}$ é o mesmo conjunto que $\{a, b\}$.
- $\{b, a, a\}$ não é o mesmo saco $\{a, b\}$.
- $\{b, a, a\}$ é o mesmo saco que $\{a, b, a\}$.
- $\{b, a, a\}$ não é a mesma lista que $\{a, b, a\}$.

SQL utiliza sacos

- SQL usa sacos (bags), álgebra relacional usa conjuntos (sets).
- Porquê sacos em vez de conjuntos?

```
SELECT nomeEstudio  
FROM Filmes;
```

```
nomeEstudio  
-----  
Fox  
Fox  
Fox  
Disney  
Paramount  
Disney  
...
```

Usar **DISTINCT** para eliminar repetidos

```
SELECT DISTINCT(nomeEstudio)
FROM Filmes;
```

nomeEstudio

Fox

Disney

Paramount

Universal

Warner Brothers

Ordenação do output

- **ORDER BY** <lista de atributos>
- ordem crescente por defeito
(colocar **DESC** para ordem decrescente)
- ordem lexicográfica para strings.

Exemplos

Filmes da Disney ordenados por ano, e dentro de cada ano, ordenados por nome.

```
SELECT *  
FROM Filmes  
WHERE nomeEstudio='Disney'  
ORDER BY ano, nome;
```

Filmes da Disney ordenados por ordem decrescente de duração, e em caso de empate, ordenados por ordem crescente de nome.

```
SELECT *  
FROM Filmes  
WHERE nomeEstudio='Disney'  
ORDER BY duracao DESC, nome;
```