

# Bases de Dados

**Módulo 18: Recursos Úteis para Exploração e Preparação de Dados**

**Prof. André Bruno de Oliveira**

17/05/24 11:36

## Recursos úteis

Este módulo cobre funcionalidades da Linguagem SQL muito úteis para os cientistas de dados executarem tarefas de exploração e preparação de dados.

## Recursos úteis

**Single-Row Functions:** também conhecidas como funções escalares, estas funções retornam um único resultado para cada linha de uma consulta.

**LIMIT:** cláusula utilizada para limitar os resultados (quantidade de linhas) retornados por um SELECT.

**Estrutura CASE / WHEN:** estrutura que funciona de forma equivalente ao CASE das linguagens de programação.

**LIKE:** operador utilizado na cláusula WHERE para permitir a busca por um padrão específico em uma coluna. Apenas padrões simples podem ser especificados.

# Recursos úteis

As seções a seguir apresentam exemplos práticos de utilização destes tipos de funcionalidades com a utilização da base de dados da “TopicosAdicionais.db”. Esta base de dados possui apenas uma tabela, que armazena informações sobre 100 filmes produzidos em diferentes épocas e países:

t\_filme (

id: INT, --chave primária (surrogate key)

titulo: VARCHAR(100), --título do filme

ano: INT, --ano de produção

resumo: VARCHAR(4000), --resumo do filme em Português

pais: CHAR(2) --sigla do país com 2 letras

duracao: INT, --duração em minutos

avaliacao: NUM --avaliação (mínimo 0.0, máximo 5.0).

)

# Recursos úteis

## 1 Funções “Single-Row”

Os principais SGBDs comerciais e livres costumam oferecer uma rica biblioteca de funções *single-row* para seus usuários.

O nome se deve ao fato deste tipo de função atuar sobre uma única linha por vez e retornar um resultado por linha (ao contrário do que ocorre com as funções de grupo - como MAX, MIN, AVG e COUNT - que atuam sobre grupos de linha e retornam um resultado por grupo).

# Recursos úteis

## 1. Funções “Single-Row”

Uma função single-row trabalha da seguinte forma:

Ela recebe um ou mais parâmetros como entrada, realiza algum processamento interno sobre esses parâmetros e retorna um valor único como saída. Os exemplos a seguir introduzem algumas das principais funções single-row oferecidas pelo SQLite.

Neste módulo, cobriremos apenas as funções de caractere (utilizadas para manipular texto), as funções numéricas (utilizadas para manipular números inteiros ou reais) e a função CAST para conversão de tipos.

# Recursos úteis

## 2 Funções de Caractere

Também chamadas de funções de string, são utilizadas para manipular colunas alfanuméricas, como CHAR, VARCHAR, TEXT, etc.

Exemplo: retorna título do filme, ano, resumo e comprimento do resumo, ordenando do maior comprimento para o menor.

```
SELECT titulo, ano, resumo, LENGTH(resumo)
```

```
FROM t_filme
```

```
ORDER BY LENGTH(resumo) DESC
```

Por questões de espaço no slid a figura mostra somente os 7 primeiros registros

	titulo	ano	resumo	LENGTH(resumo)
1	... E o Vento Levou	1939	Uma reunião social acontece numa grande plantação na...	3317
2	Coisas Belas e Sujas	2002	Em Londres um nigeriano, Okwe (Chiwetel Ejiofor), qu...	1200
3	Para o Resto de Nossas Vidas	1992	Há 10 anos atrás eles foram amigos na Universidade de...	1184
4	Os Incompreendidos	1959	Antoine Doinel (Jean-Pierre Léaud) é o filho ...	1182
5	Dança Comigo?	1996	Shohei Sugiyama (Kôji Yakusho) é um típico homem de...	977
6	O Filho da Noiva	2001	Aos 42 anos Rafael Belvedere (Ricardo Darín) está em ...	867
7	Fanny e Alexander	1982	No início do século XX, após um alegre Natal na família ...	827

# Recursos úteis

## 2.1 Funções de Caractere - LOWER e UPPER

Sintaxe: LOWER(s) e UPPER(s)

Descrição: as funções LOWER e UPPER recebem como parâmetro uma string s e retornam os seus caracteres ASCII convertidos para minúsculo e maiúsculo, respectivamente.

Exemplo: retorna o título, título em **minúsculo** e o título em **maiúsculo** dos filmes do ano de 2016:

```
SELECT titulo, LOWER(titulo), UPPER(titulo)
FROM t_filme
WHERE ano=2016
```

Observe que os caracteres acentuados não são convertidos (pois eles não fazem parte da tabela ASCII1). Felizmente, os SGBDs mais sofisticados não costumam ter esse tipo de limitação.

	titulo	LOWER(titulo)	UPPER(titulo)
1	La La Land: Cantando Estações	la la land: cantando estações	LA LA LAND: CANTANDO ESTAÇÕES



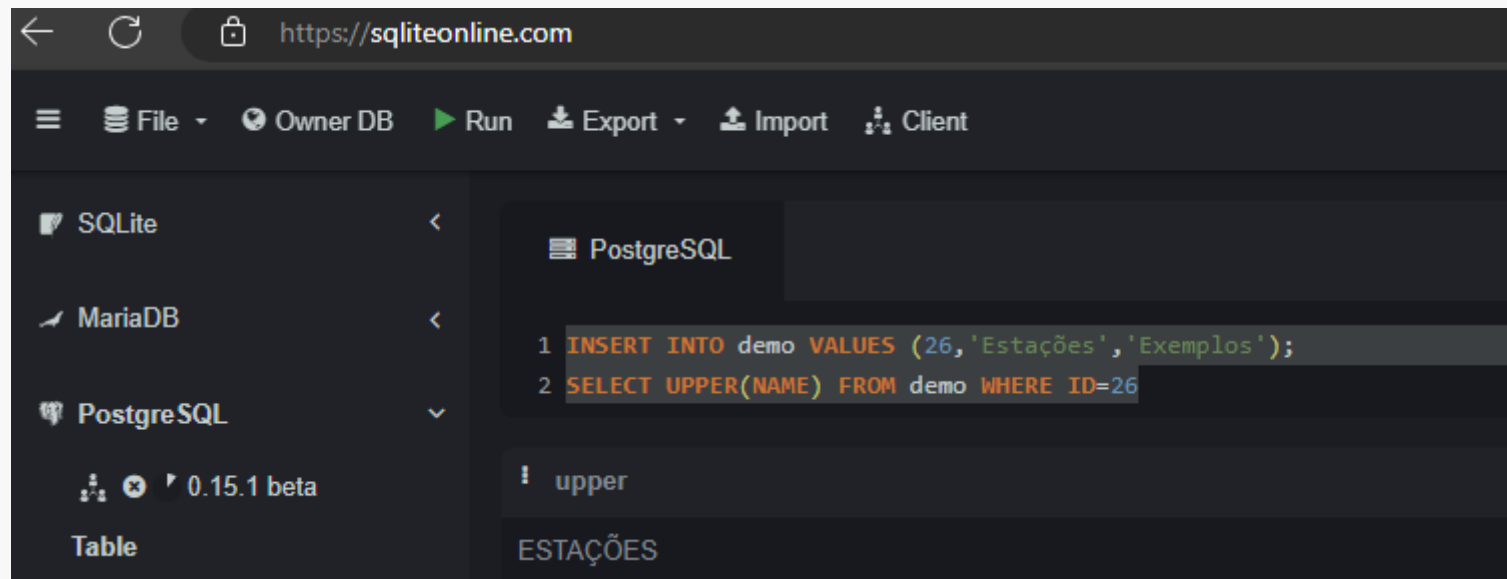
# Recursos úteis

## 2 Funções de Caractere - LOWER e UPPER

Exemplo similar feito no emulador <https://sqliteonline.com/> para o banco de dados PostgreSQL. O resultado exibe os caracteres acentuados em letras maiúsculas. O teste feito usa a própria tabela de demonstração para incluir um registro com palavra acentuada.

```
insert into demo values (26,'Estações','Exemplos');
```

```
SELECT upper(name) FROM demo where id=26
```



“Os desenvolvedores do SQLite explicam que qualquer aplicativo que precise de suporte completo a casos Unicode provavelmente já possui as tabelas e funções necessárias e, portanto, o SQLite não deve ocupar espaço para duplicar essa capacidade. Em vez de fornecer suporte completo a casos Unicode por padrão, o SQLite fornece a capacidade de vincular-se a rotinas externas de comparação e conversão de Unicode. .... ”

# Recursos úteis

## 2 Funções de Caractere - INSTR

Sintaxe: INSTR(s, sub)

Descrição: fornece a localização da substring especificada no parâmetro sub dentro da string s. Retorna 0 caso sub não faça parte de s.

Exemplo: retorna a posição da palavra “Bennet” nos resumos dos filmes de 2005.

```
SELECT titulo, resumo, INSTR(resumo, 'Bennet')  
FROM t_filme  
WHERE ano=2005
```

	titulo	resumo	INSTR(resumo, 'Bennet')
1	Orgulho e Preconceito	Inglaterra, 1797. As cinco irmãs Bennet - Elizabeth (Kei...	34
2	O que Você Faria?	Sete executivos disputam uma única vaga em uma ...	0
3	Meu Melhor Amigo	François Coste (Daniel Auteuil) é um comerciante de ...	0

# Recursos úteis

## 2 Funções de Caractere - INSTR

Sintaxe: INSTR(s, sub)

Descrição: fornece a localização da substring especificada no parâmetro sub dentro da string s. Retorna 0 caso sub não faça parte de s.

Exemplo: retorna a posição da palavra “Bennet” nos resumos dos filmes de 2005.

```
SELECT titulo, REPLACE(titulo, 'Orgulho', 'Vaidade')  
FROM t_filme  
WHERE INSTR(titulo, 'Orgulho') > 0
```

O exemplo mostra que também podemos utilizar qualquer função na cláusula WHERE, como foi feito com a função INSTR.

	titulo	REPLACE(titulo, 'Orgulho', 'Vaidade')
1	Orgulho e Preconceito	Vaidade e Preconceito

# Recursos úteis

## 2.1 Funções de Caractere - TRIM, LTRIM, RTRIM

Sintaxe: TRIM(*s*,*c*), LTRIM(*s*, *c*), RTRIM(*s*, *c*)

Descrição: A função TRIM remove todas as ocorrências do caractere *c* de ambos os lados da string *s* (caso *c* seja omitido, remove os espaços em branco). LTRIM faz a remoção apenas no lado esquerdo e RTRIM no direito.

Exemplo: remove o caractere especificado do lado de acordo com a função especificada para os filmes que começam com a string 'A Família Bél' ('A Família Bél' seguida de qualquer caractere).

```
SELECT titulo, LTRIM(titulo, 'A') , ' A Família Bélier ', LTRIM(' A Família Bélier '), RTRIM(' A Família Bélier ')
FROM t_filme
WHERE titulo LIKE 'A Família Bél%'
```

O exemplo faz respectivamente a remoção de 'A' a esquerda, remoção de branco à esquerda e de branco à direita, assim o resultado exibe as colunas modificadas. É possível combinar TRIM e LTRIM para remover esse espaço extra.

titulo	LTRIM(titulo, 'A')	' A Família Bélier '	LTRIM(' A Família Bélier ')	RTRIM(' A Família Bélier ')
1 A Família Bélier	Família Bélier	A Família Bélier	A Família Bélier	A Família Bélier

# Recursos úteis

## 2 Funções de Caractere - TRIM, LTRIM, RTRIM

Sintaxe: TRIM(*s,c*), LTRIM(*s, c*), RTRIM(*s, c*)

Descrição: Um segundo exemplo com o uso da LTRIM para remover o caractere de ambas os lados.

Exemplo: remove o caractere especificado mostra que é possível combinar as funções e o uso do TRIM para os filmes que começam com a string 'A Vida' ('A Vida' seguida de qualquer caractere).

```
SELECT titulo, TRIM(LTRIM(titulo, 'A')), ' A Vida é Bela ', TRIM(' A Vida é Bela ')  
FROM t_filme  
WHERE titulo LIKE 'A Vida%'
```

**ATENÇÃO:** A documentação do SQLite diz que TRIM pode ser usado com a especificação do caractere de remoção, só que não funciona, pelo menos nesta versão que estamos usando.

	titulo	TRIM(LTRIM(titulo, 'A'))	' A Vida é Bela '	TRIM(' A Vida é Bela ')
1	A Vida é Bela	Vida é Bela	A Vida é Bela	A Vida é Bela

# Recursos úteis

## 2 Funções de Caractere - SUBSTR

Sintaxe: SUBSTR(s, pos, tam)

Descrição: Retorna a substring de **s** que inicia na posição **pos** e tem comprimento **tam**.

Exemplo: Exibir os 30 primeiros caracteres dos resumos.

```
SELECT titulo, SUBSTR(resumo, 1, 30) FROM t_filme WHERE titulo LIKE 'A %'
```

	titulo	SUBSTR(resumo, 1, 30)
1	A Insustentáv...	Nos anos 60 em Praga, Tchecosl
2	A Copa	Orgyen é um jovem monge que te
3	A Família Bélier	Paula é uma adolescente france
4	A Vida é Bela	Durante a Segunda Guerra Mundi
5	A Onda	Em uma escola da Alemanha, alu
6	A Hora da ...	Macabéa (Marcélia Cartaxo) é u
7	A Rosa Púrpu...	Em área pobre de Nova Jersey,

# Recursos úteis

## 2 Funções de Caractere - CONCATENAÇÃO

Sintaxe: ||

Descrição: Na verdade a concatenação não é uma função, mas um operador que pode ser aplicado a valores alfanuméricos, servindo para combinar duas ou mais strings em uma só.

Exemplo: Exibir os 30 primeiros caracteres de cada resumo concatenado com '...':

```
SELECT titulo, SUBSTR(resumo, 1, 30) || '...' FROM t_filme WHERE titulo LIKE 'A %'
```

	titulo	SUBSTR(resumo, 1, 30)    '...'
1	A Insustentáv...	Nos anos 60 em Praga, Tchecosl...
2	A Copa	Orgyen é um jovem monge que te...
3	A Família Bélier	Paula é uma adolescente france...
4	A Vida é Bela	Durante a Segunda Guerra Mundi...
5	A Onda	Em uma escola da Alemanha, alu...
6	A Hora da ...	Macabéa (Marcélia Cartaxo) é u...
7	A Rosa Púrpu...	Em área pobre de Nova Jersey, ...



# Recursos úteis

## 2 Funções de Caractere - CONCATENAÇÃO

Sintaxe: ||

Descrição: Na verdade a concatenação não é uma função, mas um operador que pode ser aplicado a valores alfanuméricos, servindo para combinar duas ou mais strings em uma só.

Exemplo: Exibir os 30 primeiros caracteres de cada resumo concatenado com '...':

```
SELECT titulo, SUBSTR(resumo, 1, 30) || '...' FROM t_filme WHERE titulo LIKE 'A %'
```

	titulo	SUBSTR(resumo, 1, 30)    '...'
1	A Insustentáv...	Nos anos 60 em Praga, Tchecosl...
2	A Copa	Orgyen é um jovem monge que te...
3	A Família Bélier	Paula é uma adolescente france...
4	A Vida é Bela	Durante a Segunda Guerra Mundi...
5	A Onda	Em uma escola da Alemanha, alu...
6	A Hora da ...	Macabéa (Marcélia Cartaxo) é u...
7	A Rosa Púrpu...	Em área pobre de Nova Jersey, ...

# Recursos úteis

## 2 Funções de Conversão de Tipo - CAST

Sintaxe: `CAST(t1 AS t2)`

Descrição: Converte um valor do tipo *t1* para o tipo *t2* (ex.: de INT para FLOAT, de VARCHAR para INT, etc.).

Exemplo: Exibir a duração de um filme em horas (está originalmente armazenada em minutos).

```
SELECT titulo, ano, duracao AS dur_min, CAST(duracao AS FLOAT) / 60 AS dur_hor  
FROM t_filme ORDER BY duracao DESC
```

	titulo	ano	dur_min	dur_hor
1	... E o Vento ...	1939	238	3.966666666666667
2	Gandhi	1982	190	3.166666666666667
3	Fanny e ...	1982	188	3.133333333333333
4	Amadeus	1984	180	3
5	A Insustentáv...	1988	171	2.85
6	O Tambor	1979	162	2.7

# Recursos úteis

## 2 Funções de Conversão de Tipo - CAST

Sintaxe: CAST(*t1* AS *t2*)

Descrição: Converte um valor do tipo *t1* para o tipo *t2* (ex.: de INT para FLOAT, de VARCHAR para INT, etc.).

Exemplo: Exibir a duração de um filme em horas (está originalmente armazenada em minutos). Veja que sem a conversão só aparece o valor inteiro.

```
SELECT titulo, ano, duracao AS dur_min  
, duracao/60 as dur_hor01  
, CAST(duracao AS FLOAT) / 60 AS dur_hor  
FROM t_filme ORDER BY duracao DESC
```

	titulo	ano	dur_min	dur_hor01	dur_hor
1	... E o Vento ...	1939	238	3	3.966666666666667
2	Gandhi	1982	190	3	3.166666666666667
3	Fanny e ...	1982	188	3	3.133333333333333
4	Amadeus	1984	180	3	3
5	A Insustentáv...	1988	171	2	2.85
6	O Tambor	1979	162	2	2.7
7	Carandiru	2003	146	2	2.433333333333333

# Recursos úteis

## 2 Funções de Conversão de Tipo - CAST

Sintaxe: `CAST(t1 AS t2)`

O CAST é necessário nesse exemplo porque “duracao” é um campo INT. Por default, o SQLite retorna apenas a parte inteira da divisão sempre que o operador “/” for utilizado sobre um campo do tipo inteiro. Esse comportamento não é comum todos os SGBD (MS SQL Server por exemplo adota comportamento parecido), mas sabemos que o SQLite possui algumas peculiaridades. Veja o que acontece quando o CAST não é usado:

```
SELECT titulo, ano, duracao AS dur_min  
, duracao/60 as dur_hor01  
,CAST(duracao AS FLOAT) / 60 AS dur_hor  
FROM t_filme ORDER BY duracao DESC
```

	titulo	ano	dur_min	dur_hor01	dur_hor
1	... E o Vento ...	1939	238	3	3.966666666666667
2	Gandhi	1982	190	3	3.166666666666667
3	Fanny e ...	1982	188	3	3.133333333333333
4	Amadeus	1984	180	3	3
5	A Insustentáv...	1988	171	2	2.85
6	O Tambor	1979	162	2	2.7
7	Carandiru	2003	146	2	2.433333333333333

# Recursos úteis

## 2 Funções de Conversão de Tipo - COALESCE

Sintaxe básica: COALESCE(*x*, *valor*)

Descrição: caso o valor de *x* for NULL, será trocado por *valor*.

Exemplo: o exemplo abaixo trocará NULL por 0, sempre que o valor de número for NULL.

```
SELECT COALESCE(numero, 0) FROM  
(SELECT 1000 AS numero  
UNION  
SELECT NULL AS numero  
UNION SELECT 2000 AS numero )  
t ORDER BY numero
```

	COALESCE(numero, 0)
1	0
2	1000
3	2000

# Recursos úteis

## 3 Funções Numéricas - ROUND

Sintaxe: ROUND ( $n,d$ )

Descrição: Arredonda o número  $n$  para um certo número de casas decimais, especificado no parâmetro  $d$ . Caso  $d$  seja omitido, arredonda para o inteiro mais próximo (se o valor da parte decimal é maior ou igual a .5, arredonda pra cima; senão, arredonda para baixo).

Exemplo: Exibir a duração de um filme em horas, com duas casas decimais

```
SELECT titulo, ano, duracao AS dur_min,  
       ROUND(CAST(duracao AS FLOAT) / 60,2) AS dur_hor  
FROM t_filme ORDER BY duracao DESC
```

	titulo	ano	dur_min	dur_hor
1	... E o Vento ...	1939	238	3.97
2	Gandhi	1982	190	3.17
3	Fanny e ...	1982	188	3.13
4	Amadeus	1984	180	3
5	A Insustentáv...	1988	171	2.85
6	O Tambor	1979	162	2.7
7	Carandiru	2003	146	2.43
8	Julietta dos ...	1965	137	2.28

# Recursos úteis

## 3 Funções Numéricas – RANDOM

Sintaxe: RANDOM()

Descrição: Retorna um valor aleatório entre -9223372036854775808 e +9223372036854775807.

```
SELECT RANDOM()
```

	RANDOM()
1	2900474367699865783

Veja que neste exemplo, fizemos uso do SELECT sem FROM! Isso é permitido no SQLite e em alguns outros SGBDs, em situações onde deseja-se obter um resultado que não vem de nenhuma tabela.

# Recursos úteis

## 3 Funções Numéricas – ABS

Sintaxe: ABS()

Descrição: Retorna o valor absoluto (módulo) de um número.

Exemplo: Exibir a duração de um filme em horas, com duas casas decimais

```
SELECT ABS(1000), ABS(-500), ABS(2 * -2.5), ABS(0)
```

	ABS(1000)	ABS(- 500)	ABS(2 * - 2.5)	ABS(0)
1	1000	500	5	0



# Recursos úteis

## 3.4 Exemplos Finais

A seguir um exemplo onde diferentes funções são utilizadas nas cláusulas SELECT, FROM e ORDER BY. O exemplo lista o título, ano, resumo e comprimento do resumo de todos os filmes que possuem 'Rio de Janeiro' no resumo (independente de estar escrita em letras maiúsculas ou minúsculas), ordenando os resultados pelo comprimento do resumo.

```
SELECT titulo, ano, resumo, LENGTH(resumo) AS compr_resumo  
FROM t_filme  
WHERE INSTR(UPPER(resumo),'RIO DE JANEIRO') > 0  
ORDER BY LENGTH(resumo);
```

	titulo	ano	resumo	compr_resumo
1	Central do ...	1998	Dora (Fernanda Montenegro) trabalha escrevendo carta...	407
2	Onde Está ...	2018	Inspirado no livro HO-BA-LA-LÁ - À Procura de João ...	447
3	Se Segura, ...	1978	Rio de Janeiro. Paulo Otávio (Hugo Carvana) mantém ...	479
4	Nise - O ...	2015	Ao voltar a trabalhar em um hospital psiquiátrico no ...	486
5	Elis	2015	Cantora desde a infância, Elis Regina Carvalho Costa ...	503
6	Leila Diniz	1987	Biografia da artista Leila Diniz (Louise Cardoso), nascida...	558
7	Gonzaga: de ...	2012	Decidido a mudar seu destino, Gonzaga sai de casa ...	820

# Recursos úteis

## 3.4 Exemplos Finais

O exemplo a seguir combina o uso de uma função de grupo com uma função single-row.

O exemplo mostra a média das avaliações e o total de filmes agrupado por país, utilizando ROUND na média para arredondar os resultados para uma casa decimal:

```
SELECT pais, ROUND(AVG(avaliacao),1) AS nota_media, COUNT(*) AS tot_filmes
```

```
FROM t_filme
```

```
GROUP BY pais
```

```
ORDER BY 2 DESC, 1;
```

	pais	nota_media	tot_filmes
1	CL	4.2	1
2	UK	4.1	18
3	AR	4	5
4	IT	4	3
5	US	4	17
6	UY	4	2
7	CA	3.9	2
8	FR	3.9	10
9	BR	3.8	24
10	DE	3.8	5
11	ES	3.8	5
12	IN	3.8	1
13	SE	3.8	2
14	BE	3.7	1
15	JP	3.4	1
16	BT	3.2	1
17	CU	3.2	1
18	KZ	3.1	1

# Recursos úteis

## 3.5 Exemplos Finais - LIMIT

Esta cláusula é utilizada para limitar a quantidade de resultados retornados por um comando SELECT. Sua sintaxe básica é apresentada abaixo. O WHERE e ORDER BY são opcionais. Veja que o LIMIT deve ser especificado após o ORDER BY.

Sintaxe:

SELECT colunas

FROM tabela

[WHERE]

[ORDER BY]

LIMIT número de linhas

# Recursos úteis

## 3.5 Exemplos Finais - LIMIT

Exemplo: Retornar apenas os 8 primeiros filmes produzidos entre 1970 e 1979.

Sintaxe:

SELECT titulo, ano

FROM t\_filme

WHERE ano BETWEEN 1970 AND 1979

ORDER BY ano

LIMIT 8;

	titulo	ano
1	Ensina-me a Viver	1971
2	O Discreto Charme da Burguesia	1972
3	Monty Python em Busca do Cálice Sagrado	1975
4	Se Segura, Malandro!	1978
5	Bye Bye Brasil	1979
6	Hair	1979
7	Mamãe Faz 100 Anos	1979
8	O Casamento de Maria Braun	1979

# Recursos úteis

## 3.5 Exemplos Finais - LIMIT

A cláusula OFFSET pode ser utilizada para determinar qual será o primeiro resultado apresentado por LIMIT.

Sintaxe:

SELECT colunas

FROM tabela

[WHERE]

[ORDER BY]

LIMIT número de linhas OFFSET valor

# Recursos úteis

## 3.5 Exemplos Finais - LIMIT

**Exemplo:** Retornar apenas os 8 primeiros filmes produzidos entre 1970 e 1979, iniciando a partir da quarta linha (não mostrará mais “Ensina-me a Viver”, “Estado de Sítio” e “O Discreto Charme da Burguesia” - o uso de OFFSET 3 faz com que 3 linhas sejam puladas).

```
SELECT titulo, ano
```

```
FROM t_filme
```

```
WHERE ano BETWEEN 1970 AND 1979
```

```
ORDER BY ano LIMIT 8 OFFSET 3
```

	titulo	ano
1	Se Segura, Malandro!	1978
2	Bye Bye Brasil	1979
3	Hair	1979
4	Mamãe Faz 100 Anos	1979
5	O Casamento de Maria Braun	1979
6	O Tambor	1979

# Recursos úteis

## 3.6 Exemplos Finais - CASE WHEN

**Exemplo:** Retornar apenas os 8 primeiros filmes produzidos entre 1970 e 1979, iniciando a partir da quarta linha (não mostrará mais “Ensina-me a Viver”, “Estado de Sítio” e “O Discreto Charme da Burguesia” - o uso de OFFSET 3 faz com que 3 linhas sejam puladas).

Sintaxe:

CASE [expressão]

WHEN condição\_1 THEN resultado\_1

WHEN condição\_2 THEN resultado\_2

...

WHEN condição\_n THEN resultado\_n

ELSE resultado

END

# Recursos úteis - CASE WHEN

Esta estrutura é especialmente útil quando desejamos converter valores numéricos em categorias, como mostra o exemplo a seguir.

**Exemplo:** retorna o conceito (A+, A, A- ou B) de um filme, que é determinado em função de sua nota de avaliação.

```
SELECT titulo, ano, avaliacao,
```

```
CASE
```

```
WHEN avaliacao BETWEEN 4.4 AND 5.0 THEN 'A+'
```

```
WHEN avaliacao BETWEEN 4.0 AND 4.3 THEN 'A'
```

```
WHEN avaliacao BETWEEN 3.5 AND 3.9 THEN 'A-'
```

```
ELSE 'B'
```

```
END AS conceito
```

```
FROM t_filme ORDER BY titulo;
```

	titulo	ano	avaliacao	conceito
1	... E o Vento Levou	1939	4.5	A+
2	A Copa	1999	3.2	B
3	A Família Bélier	2014	4.4	A+
4	A Hora da Estrela	1985	3.6	A-
5	A Insustentável Leveza do Ser	1988	3.9	A-
6	A Onda	2008	4.5	A+
7	A Rosa Púrpura do Cairo	1985	4	A



# Recursos úteis - CASE WHEN

## 3.7 Operador LIKE

O operador LIKE é utilizado para “casar” valores texto contra um padrão que é determinado através das chamadas *wildcards*. O LIKE é bastante simples de usar, pois existem apenas duas *wildcards*:

- % : O sinal de percentual representa múltiplos caracteres
- \_ : O *underscore* representa um único caractere

Apesar de sua simplicidade (ou talvez por causa dela!), o LIKE é um comando muito utilizado e que funciona bem para um grande número de situações práticas.

Exemplo: retornar todos os filmes cujo título começa com M:

```
SELECT titulo, ano FROM t_filme WHERE titulo LIKE 'M%'
```

	titulo	ano
1	Macunaíma	1969
2	Mamãe Faz 100 Anos	1979
3	Meu Melhor Amigo	2005
4	Monty Python em Busca do Cálice Sagrado	1975
5	Morango e Chocolate	1993

# Recursos úteis - CASE WHEN

## 3.7 Operador LIKE

Exemplo: retornar todos os filmes cujo título termina com 'as'.

```
SELECT titulo, ano FROM t_filme WHERE titulo LIKE 'M%'
```

	titulo	ano
1	As Invasões Bárbaras	2003
2	Coisas Belas e Sujas	2002
3	Nove Rainhas	2000
4	Para o Resto de Nossas Vidas	1992
5	Peixe Grande e suas Histórias Maravilhosas	2003

# Recursos úteis - CASE WHEN

## 3.7 Operador LIKE

Exemplo: retornar todos os filmes que possuem 'RIO DE JANEIRO' em qualquer posição do resumo.

```
SELECT titulo, ano FROM t_filme
```

```
WHERE UPPER(resumo) LIKE '%RIO DE JANEIRO%'
```

	titulo	ano
1	Nise - O Coração da Loucura	2015
2	Se Segura, Malandro!	1978
3	Leila Diniz	1987
4	Elis	2015
5	Central do Brasil	1998
6	Gonzaga: de Pai para Filho	2012
7	Onde Está Você, João Gilberto?	2018

# Recursos úteis - CASE WHEN

## 3.7 Operador LIKE

Exemplo: retornar todos os filmes que possuem a letra “u” na segunda posição do título.

```
SELECT titulo, ano
```

```
FROM t_filme
```

```
WHERE titulo LIKE '_u%'
```

	titulo	ano
2	Julieta dos Espíritos	1965
3	Lunchbox	2013
4	Qual é o Nome do Bebê?	2012
5	Que Horas Ela Volta?	2015
6	Quem Quer Ser um Milionário?	2008
7	Tudo Sobre Minha Mãe	1999
8	Tulpan	2010

# Recursos úteis - CASE WHEN

## 3.7 Operador LIKE

Observação: Ao contrário do que ocorre quando usamos as funções de caractere, o SQLite não faz distinção entre caracteres maiúsculos e minúsculos no LIKE (algo diferente do que ocorre na maioria dos outros SGBDs). Isto quer dizer que se você fizer LIKE '%A' ou LIKE '%a' ele achará os mesmos resultados! Para modificar esse comportamento é preciso alterar o valor do parâmetro “case\_sensitive\_like”.

--LIKE passa a distinguir caracteres minúsculos e maiúsculos

PRAGMA case\_sensitive\_like=ON;

--LIKE passa a não distinguir caracteres minúsculos e maiúsculos

PRAGMA case\_sensitive\_like=OFF;

# Recursos úteis - CASE WHEN

## EXPRESSÕES REGULARES

De uma maneira simplificada, podemos dizer que as expressões regulares possuem o mesmo propósito do operador LIKE: permitir a especificação de um padrão para ser “casado” com valores de um texto. No entanto, as expressões regulares possibilitam com que padrões muito mais complexos possam ser definidos. Trata-se de uma tecnologia tão popular e importante que está disponível não apenas na SQL, mas praticamente em todas as linguagens de programação, incluindo R e Python.

# Recursos úteis - CASE WHEN

## EXPRESSÕES REGULARES

No campo da Estatística, as expressões regulares são especialmente úteis para aqueles que desejam trabalhar em projetos envolvendo Mineração de Texto e *Web Scraping* (Solução adota para coletar dados de sites públicos de Internet através da busca de padrões). O SQLite suporta o uso de expressões regulares através do operador REGEXP. Apenas para você ter uma ideia, o SELECT abaixo retorna todos os filmes que possuem algum número aparecendo em seu título (seja qual for o número!):

```
SELECT titulo, ano FROM t_filme WHERE titulo REGEXP '[0-9]'
```

	titulo	ano
1	Fahrenheit 451	1966
2	Mamãe Faz 100 Anos	1979
3	Uma Noite de 12 Anos	2018

Para conhecer sobre o assunto, recomendamos o excelente tutorial disponibilizado em:  
<http://aurelio.net/regex/guia/>.

Obrigado