

Cláusula WHERE

Banco de Dados

Charles Tim Batista Garrocho

Instituto Federal de Minas Gerais – IFMG
Campus Ponte Nova

`garrocho.github.io/BDD`

`charles.garrocho@ifmg.edu.br`

Técnico em Informática



Claúsula WHERE

A cláusula **WHERE** é uma cláusula opcional da instrução SELECT. Aparece após a cláusula FROM como a seguinte declaração:

```
SELECT * FROM TABELA WHERE CONDICoes_PESQUISA;
```

Você adiciona uma cláusula WHERE à instrução SELECT para **filtrar** os dados retornados pela consulta.

A cláusula WHERE também é conhecida como um **conjunto de condições** ou uma lista de predicados.



Operadores de comparação SQLite

Um **operador de comparação** testa se duas expressões são **iguais**.

A tabela a seguir ilustra os operadores de comparação que você pode usar para construir expressões.

Operador	Significado
=	Igual a
<> ou !=	Diferente que
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a



Operadores lógicos SQLite

Os operadores lógicos permitem **testar a verdade** de algumas expressões. Um operador lógico retorna 1, 0 ou um valor NULL. A tabela a seguir ilustra os operadores lógicos do SQLite.

Operador	Significado
AND	retorna 1 se ambas as expressões forem 1 e 0 se uma das expressões for 0
BETWEEN	retorna 1 se um valor estiver dentro de um intervalo
IN	retorna 1 se um valor estiver em uma lista de valores
LIKE	retorna 1 se um valor corresponder a um padrão
NOT	inverte o valor de outros operadores, como: NOT EXISTS, NOT IN, NOT BET, etc
OR	retorna verdadeiro se uma das expressões for 1



Claúsula WHERE: Operador Lógico AND

Considerando a tabela Notas abaixo, o comando SQL para selecionar os alunos que conseguiram notas maiores que 8 em História e Geografia é:

```
SELECT ALUNO FROM NOTAS  
WHERE HISTORIA > 8 AND GEOGRAFIA > 8;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Adriana

Fernanda



Claúsula WHERE: Operador Lógico OR

Considerando a tabela Notas abaixo, o comando SQL para selecionar os alunos que tiraram nota 7 ou 10 em Português é:

```
SELECT ALUNO FROM NOTAS WHERE PORTUGUES = 4 OR  
PORTUGUES = 10;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Adriana

Joãozinho



Claúsula WHERE: Operador Lógico IN

Considerando a tabela Notas abaixo, o comando SQL para selecionar os alunos que tiraram nota 2 ou 4 ou 6 em Português é:

```
SELECT ALUNO FROM NOTAS WHERE PORTUGUES IN (2, 4, 6);
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:
Joãozinho



Claúsula WHERE: Operador Lógico BETWEEN

Considerando a tabela Notas abaixo, o comando SQL para selecionar os alunos que conseguiram notas entre 9 e 10 em Matemática é:

```
SELECT ALUNO FROM NOTAS  
WHERE MATEMÁTICA BETWEEN 9 AND 10;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Adriana

Carlos



Claúsula WHERE: Operador Lógico LIKE

Considerando a tabela Notas abaixo, o comando SQL para selecionar os alunos cujo nome comece com a letra 'A' é:

```
SELECT ALUNO FROM NOTAS WHERE ALUNO LIKE 'A%';
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Adriana

Alexandre



Claúsula WHERE: Operador Lógico NOT

Considerando a tabela Notas abaixo, o comando SQL para selecionar os alunos cujo nome não comece com a letra 'A' é:

```
SELECT ALUNO FROM NOTAS  
WHERE ALUNO NOT LIKE 'A%';
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Carlos

Fernanda

Joãozinho



Atividades Práticas

Utilizando o Shell SQLite3, implemente uma base de dados para gerenciar os pacientes de um médico. Esta base de dados permitirá armazenar o nome, telefone e idade de cada paciente. Após isso, mostre os comandos necessários para:

- Inserir 6 pacientes;
- Imprimir os pacientes com idade entre 10 e 15 anos;
- Imprimir os pacientes maiores de 18 anos;
- Imprimir os pacientes maiores de 18 anos e o nome contenha "silva";
- Imprimir os pacientes menores de idade e com mais de 50 anos;
- Imprimir os pacientes que tenham 10, 20, 30 ou 40 anos;
- Imprimir os pacientes que possuem o sobrenome "da silva";
- Imprimir os pacientes que possuem o telefone com DDD 31;
- Imprimir os pacientes que não possuem o telefone com DDD 31;

Obs: Guarde os comandos para a análise do professor.

