Vitor Amadeu Souza

Introdução ao

SQLite

Volume VI

© 2024 by Cerne Tecnologia e Treinamento Ltda.

© 2024 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Julho de 2024

Direitos reservados por:
Cerne Tecnologia e Treinamento Ltda
Produção: Cerne Tecnologia e Treinamento
E-mail da Empresa: cerne@cerne-tec.com.br
Home Page: www.cerne-tec.com.br.com.br
Atendimento ao Consumidor: sac@cerne-tec.com.br
Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

“Recompensou-me o Senhor conforme a minha justiça,

retribuiu-me conforme a pureza das minhas mãos.”

Sl 18:20

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM, STM etc. Veja os detalhes de um kit na figura abaixo:



Kit Cerne PIC12F

- Microcontrolador PIC12F675
- Botões
- Leds
- Gravação ICSP
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais dentre outros. Visite a nossa página na Internet, no endereço www.cerne-tec.com.br e conheça melhor nossos serviços e produtos.

Cerne
Conhecimento para o Desenvolvimento

Sumário

Introdução.....6

Capítulo I – SQLite.....7

1. RIGHT JOIN.....7
2. FULL JOIN.....9
3. ALTER TABLE.....11
4. DROP TABLE.....15
5. CREATE VIEW.....16
6. CONCAT.....18
7. IS NULL.....19

Introdução

A proposta desta literatura é apresentar o acesso ao banco de dados SQLite usando para isso a plataforma on-line Replit. Em cada parte desta série de livros uma parte do SQLite será apresentada, no qual o tema abordado está no sumário de cada obra.

Em todos os casos, sempre que possível, exemplos práticos escritos em Python procuram envolver a base de dados com programação, facilitando desta forma a compreensão deste assunto.

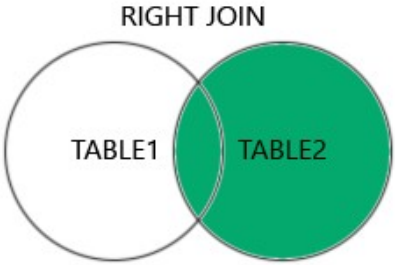
Capítulo I

SQLite

1. RIGHT JOIN

O RIGHT JOIN é um tipo de junção utilizado em consultas SQL que retorna todas as linhas da tabela à direita da junção (tabela B) e as linhas correspondentes da tabela à esquerda da junção (tabela A). Se não houver correspondência, os resultados ainda incluirão todas as linhas da tabela B, mas com NULL para as colunas da tabela A onde não há correspondência.

Fazendo a representação em diagrama de Venn, observamos que o RIGHT JOIN pode ser representado segundo a figura a seguir.



Fonte: https://www.w3schools.com/sql/sql_join_right.asp

O próximo exemplo demonstra o uso do RIGHT JOIN, onde nem todos os pedidos foram feitos por clientes cadastrados , entretanto ao se usar o RIGHT JOIN terão a sua correspondência apresentada.

```
CREATE TABLE Clientes
(
   ClienteID INT PRIMARY KEY,
   Nome VARCHAR(100)
);
CREATE TABLE Pedidos
(
   PedidoID INT PRIMARY KEY,
   ClienteID INT,
   Produto VARCHAR(100)
);
INSERT INTO Clientes (ClienteID, Nome) VALUES
    (1, 'Alice'),
    (2, 'Bob'),
    (3, 'Charlie');
INSERT INTO Pedidos (PedidoID, ClienteID, Produto) VALUES
    (101, 2, 'Produto A'),
    (102, 3, 'Produto B'),
    (103, 4, 'Produto C');
```

```
SELECT Clientes.ClienteID, Clientes.Nome, Pedidos.PedidoID, Pedidos.Produto
FROM Clientes
RIGHT JOIN Pedidos
ON Clientes.ClienteID = Pedidos.ClienteID;
```

```
-- Loading resources from main.sql
2|Bob|101|Produto A
3|Charlie|102|Produto B
||103|Produto C
SQLite version 3.43.2 2023-10-10 12:14:04
Enter ".help" for usage hints.
sqlite>
```

Note que foi inserido um cliente não cadastrado, porém usando o RIGHT JOIN o pedido é apresentado com o valor NULL.

2. FULL JOIN

O FULL JOIN (também conhecido como FULL OUTER JOIN) é um tipo de junção em SQL que retorna todas as linhas quando há uma correspondência em uma das tabelas. Ele combina os resultados de um LEFT JOIN e um RIGHT JOIN. Se não houver correspondência em uma das tabelas, os resultados ainda incluirão todas as linhas das duas tabelas, mas com NULL nas colunas da tabela onde não houve correspondência.

O próximo exemplo demonstra o uso do FULL JOIN.

```
CREATE TABLE Clientes (
   ClienteID INT PRIMARY KEY,
   Nome VARCHAR(100)
);

CREATE TABLE Pedidos (
   PedidoID INT PRIMARY KEY,
   ClienteID INT,
   Produto VARCHAR(100)
);

INSERT INTO Clientes (ClienteID, Nome) VALUES
    (1, 'Alice'),
    (2, 'Bob'),
    (3, 'Charlie');

INSERT INTO Pedidos (PedidoID, ClienteID, Produto) VALUES
    (101, 2, 'Produto A'),
    (102, 3, 'Produto B'),
    (103, 4, 'Produto C');

SELECT Clientes.ClienteID, Clientes.Nome, Pedidos.PedidoID, Pedidos.Produto
FROM Clientes
```