Vitor Amadeu Souza

Introdução a

SQLite

Com exemplos práticos em Python

Volume III

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

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, exemplos práticos escritos em Python procuram envolver a base de dados com programação, facilitando desta forma a compreensão deste assunto.

Capítulo I

SQLite

1. Chave primária

No SQL, uma chave primária é um campo (ou conjunto de campos) em uma tabela que identifica de maneira única cada registro dessa tabela. As características principais de uma chave primária são:

1. Unicidade: Cada valor da chave primária deve ser único em toda a tabela, garantindo que nenhum registro seja idêntico a outro em termos desse(s) campo(s).

2. Não nulo: Valores da chave primária não podem ser nulos, pois ela deve identificar de maneira única cada registro.

3. Imutabilidade: Embora não seja uma regra obrigatória em todos os sistemas de gerenciamento de bancos de dados (SGBDs), geralmente é uma boa prática que os valores da chave primária não mudem ao longo do tempo.

A chave primária é utilizada para criar índices eficientes que ajudam nas operações de busca e na manutenção da integridade referencial dos dados. A seguir um exemplo básico de como definir uma chave primária em uma tabela SQL:

```
CREATE TABLE Clientes
(
        ID INT NOT NULL,
        Nome VARCHAR(100),
        Email VARCHAR(100),
        PRIMARY KEY (ID)
);
```

Neste exemplo, o campo `ID` é definido como a chave primária da tabela `Clientes`. Isso significa que cada valor em `ID` deve ser único e não nulo.

Como exemplo, elabore uma tabela chamada alunos com os campos: ID (PK), nome, endereço, telefone, nascimento, CPF, RG, nome do Pai e Mãe. Insira alguns dados e depois selecione tudo e apresente os mesmos.

```
CREATE TABLE alunos (
   ID INT PRIMARY KEY,
   nome VARCHAR(100) NOT NULL,
   endereco VARCHAR(255),
   telefone VARCHAR(20),
   nascimento DATE,
   CPF VARCHAR(11) UNIQUE NOT NULL,
   RG VARCHAR(20) UNIQUE NOT NULL,
   nome_pai VARCHAR(100),
   nome_mae VARCHAR(100)
);

INSERT INTO alunos (ID, nome, endereco, telefone, nascimento, CPF, RG, nome_pai,
nome_mae)
VALUES
(1,'João Silva', 'Rua A, 123', '1111-2222', '2000-01-01', '12345678901', 'MG1234567',
'Carlos Silva', 'Maria Silva'),
(2,'Ana Pereira', 'Rua B, 456', '3333-4444', '1998-05-15', '23456789012', 'SP2345678',
'José Pereira', 'Clara Pereira'),
(3,'Pedro Souza', 'Rua C, 789', '5555-6666', '2002-07-20', '34567890123', 'RJ3456789',
'Marcos Souza', 'Fernanda Souza');


SELECT * FROM alunos;
```

```
-- Loading resources from main.sql
1|João Silva|Rua A, 123|1111-2222|2000-01-01|12345678901|MG1234567|Carlos Silva|Maria Silva
2|Ana Pereira|Rua B, 456|3333-4444|1998-05-15|23456789012|SP2345678|José Pereira|Clara Pereira
3|Pedro Souza|Rua C, 789|5555-6666|2002-07-20|34567890123|RJ3456789|Marcos Souza|Fernanda Souza
SQLite version 3.43.2 2023-10-10 12:14:04
Enter ".help" for usage hints.
sqlite>
```

2. ORDER BY

A cláusula ORDER BY no SQL é utilizada para ordenar o resultado de uma consulta em ordem crescente ou decrescente, com base em um ou mais campos. Por padrão, a ordenação é feita em ordem crescente (ASC), mas também pode ser especificada a ordem decrescente (DESC).

Como exemplo, observe o exemplo abaixo que elabora a tabela alunos com 4 campos, insere dados e em seguida lista os mesmos ordenando pelo nome tal impressão.

```
CREATE TABLE alunos
(
ID INT PRIMARY KEY,
nome VARCHAR(100) NOT NULL,
nascimento DATE,
telefone VARCHAR(20)
);
```