

SEMANA 08 - TI - Avançado

Aluno: Jhonatan Thiago Pereira do Nascimento

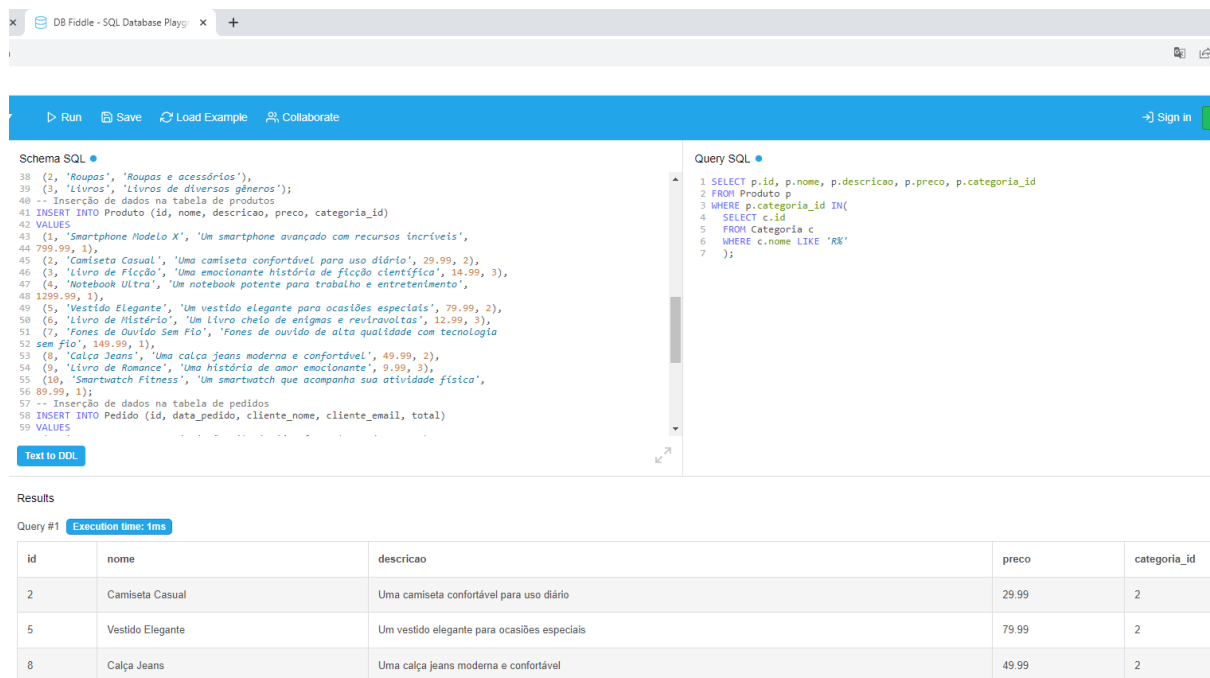
04/10/2023

Banco de dados

Aula 13 Linguagem SQL – Subconsultas

EXERCÍCIO 1

Crie as instruções SQL que liste todos os produtos que estão em uma categoria cujo o nome inicie com a letra R. Utilize subconsultas.



The screenshot shows the DB Fiddle interface with a SQL query and its results. The query is a SELECT statement using a subquery to filter products by category name starting with 'R'.

Schema SQL

```
38 (2, 'Roupas', 'Roupas e acessórios');
39 (3, 'Livros', 'Livros de diversos gêneros');
40 -- Inserção de dados na tabela de produtos
41 INSERT INTO Produto (id, nome, descricao, preco, categoria_id)
42 VALUES
43 (1, 'Smartphone Modelo X', 'Um smartphone avançado com recursos incríveis',
44 799.99, 1),
45 (2, 'Camiseta Casual', 'Uma camiseta confortável para uso diário', 29.99, 2),
46 (3, 'Livro de Ficção', 'Uma emocionante história de ficção científica', 14.99, 3),
47 (4, 'Notebook Ultra', 'Um notebook potente para trabalho e entretenimento',
48 1299.99, 1),
49 (5, 'Vestido Elegante', 'Um vestido elegante para ocasiões especiais', 79.99, 2),
50 (6, 'Livro de Mistério', 'Um livro cheio de enigmas e reviravoltas', 12.99, 3),
51 (7, 'Fones de Ouvido Sem Fio', 'Fones de ouvido de alta qualidade com tecnologia
52 sem fio', 149.99, 1),
53 (8, 'Calça Jeans', 'Uma calça jeans moderna e confortável', 49.99, 2),
54 (9, 'Livro de Romance', 'Uma história de amor emocionante', 9.99, 3),
55 (10, 'Smartwatch Fitness', 'Um smartwatch que acompanha sua atividade física',
56 89.99, 1);
57 -- Inserção de dados na tabela de pedidos
58 INSERT INTO Pedido (id, data_pedido, cliente_nome, cliente_email, total)
59 VALUES
```

Query SQL

```
1 SELECT p.id, p.nome, p.descricao, p.preco, p.categoria_id
2 FROM Produto p
3 WHERE p.categoria_id IN(
4 SELECT c.id
5 FROM Categoria c
6 WHERE c.nome LIKE 'R%')
7 );
```

Results

Query #1 Execution time: 1ms

id	nome	descricao	preco	categoria_id
2	Camiseta Casual	Uma camiseta confortável para uso diário	29.99	2
5	Vestido Elegante	Um vestido elegante para ocasiões especiais	79.99	2
8	Calça Jeans	Uma calça jeans moderna e confortável	49.99	2

<https://www.db-fiddle.com/f/6owof9mvusdAHXtu7JgJNY/0>

EXERCÍCIO 2

Crie uma consulta SQL traga uma lista com o nome dos produtos, seu respectivo preço e a quantidade de itens vendidos em ordem decrescente de quantidade de vendas. Apenas produtos vendidos devem aparecer na lista.

DB Fiddle - Playground de banco de dados

v/f/6owof9mvusdAHXtu7JgJNY/0

v 8.0 | Correr | Atualizar | Garfo | Exemplo de carga | Estrela | PRÓ | < > | Embutir | PRÓ | Colaborar | Entrar

Esquema SQL

```
1 -- Criação da tabela de categorias
2 CREATE TABLE Categoria (
3   id INT PRIMARY KEY,
4   nome VARCHAR(255) NOT NULL,
5   descricao TEXT
6 );
7 -- Criação da tabela de produtos
8 CREATE TABLE Produto (
9   id INT PRIMARY KEY,
10  nome VARCHAR(255) NOT NULL,
11  descricao TEXT,
12  preco DECIMAL(10, 2) NOT NULL,
13  categoria_id INT,
14  FOREIGN KEY (categoria_id) REFERENCES Categoria(id)
15 );
16 -- Criação da tabela de pedidos
17 CREATE TABLE Pedido (
18   id INT PRIMARY KEY,
19   data_pedido DATETIME NOT NULL,
20   cliente_nome VARCHAR(255) NOT NULL,
21   cliente_email VARCHAR(255) NOT NULL,
22   total DECIMAL(10, 2) NOT NULL
23 );
```

Consultar SQL

```
1 SELECT p.nome AS nome, p.preco AS preco, SUM(ip.quantidade) AS quantidade
2 FROM Produto p
3 JOIN ItemPedido ip ON p.id = ip.produto_id
4 GROUP BY p.id, p.nome, p.preco
5 HAVING quantidade > 0
6 ORDER BY quantidade DESC;
```

Resultados

Consulta nº 1 | Tempo de execução: 1ms

nome	barato	quantidade
Livro de Romance	9,99	3
Livro de Ficção	14,99	2
Smartphone Modelo X	799,99	1
Vestido Elegante	79,99	1

<https://www.db-fiddle.com/f/6owof9mvusdAHXtu7JgJNY/1>

Desenvolvimento Backend

Aula 13 Autenticando usuários na API

EXERCÍCIO 1

Crie um programa em Node.js que gere tokens JWT que satisfaça os seguintes requisitos:

- A token deve ser obtida a partir de uma rota /token;
- A secret deve ser obtida a partir da leitura de uma variável de ambiente;
- A token deve possuir tempo de expiração de 1 hora;
- A token deve ter um payload com um número aleatório gerado entre 0 e 1.

RESPONDIDO EM UM ARQUIVO EXTERNO

Aula 14 Documentando uma API

EXERCÍCIO 1

Baixe o projeto de uma API simples em Node.js no link abaixo e crie uma documentação seguindo a especificação do OpenAPI. Analise o código no arquivo index.js para descobrir as rotas e também o arquivo que realiza as chamadas da API (api.http).

RESPONDIDO EM UM ARQUIVO EXTERNO