

FUNÇÕES E PROCEDIMENTOS EM SQL

MÓDULO I

FUNÇÕES

As funções em SQL são como ferramentas mágicas que nos ajudam a manipular e trabalhar com nossos dados de maneira mais eficiente. Elas vêm em diferentes formas e tamanhos, cada uma com sua própria habilidade especial. Vamos dar uma olhada em alguns tipos de funções e como elas podem ser úteis.

Imagine que você tem uma tabela cheia de números e precisa encontrar a soma de todos eles. Aqui é onde entram as funções de agregação, como SUM. Com apenas uma linha de código, você pode somar todos os valores em uma coluna e obter o resultado desejado.

```
SELECT SUM(valor) AS total FROM minha_tabela;
```

Agora, suponha que você tenha uma coluna com nomes de produtos e precisasse juntar o nome do produto com uma descrição adicional. É aqui que as funções de manipulação de strings entram em jogo. Você pode usar a função CONCAT para combinar duas strings em uma só.

```
SELECT CONCAT(nome_produto, ' - ', descricao) AS produto_completo FROM produtos;
```

E se você precisar descobrir quantos dias se passaram desde uma determinada data? As funções de data/hora, como DATEDIFF, podem resolver isso para você. Com apenas uma linha de código, você pode calcular a diferença entre duas datas.

```
SELECT DATEDIFF(NOW(), data_inicial) AS dias_passados FROM minha_tabela;
```

Por último, imagine que você precisa arredondar um número para o inteiro mais próximo. As funções matemáticas, como ROUND, podem fazer isso por você. Com apenas uma linha de código, você pode arredondar um número para o número inteiro mais próximo.

```
SELECT ROUND(valor) AS valor_arredondado FROM minha_tabela;
```

ATENÇÃO

Além desses exemplos, também podemos criar nossas próprias funções personalizadas para realizar tarefas específicas que não são cobertas pelas funções embutidas. Compreender e dominar o uso dessas funções nos permite realizar consultas avançadas e análises de dados eficazes em SQL. Meu caro aluno! É claro que não podemos esquecer das funções criadas pelo usuário! Essas funções são como poções mágicas que você mesmo cria para resolver problemas específicos que não podem ser resolvidos com as funções embutidas.

Funções escalares e funções de tabela

Existem dois tipos principais de funções criadas pelo usuário em SQL: funções escalares e funções de tabela.

Funções escalares

Essas funções aceitam um ou mais parâmetros e retornam um único valor. Elas são ótimas para encapsular lógica de negócios complexa ou realizar cálculos personalizados. Por exemplo, imagine que você precise de uma função para calcular a idade de uma pessoa com base em sua data de nascimento. Você pode criar uma função escalar chamada CALCULAR_IDADE que aceita a data de nascimento como entrada e retorna a idade calculada.

```
CREATE FUNCTION CALCULAR_IDADE (data_nascimento DATE)
RETURNS INT
BEGIN
DECLARE idade INT;
SET idade = YEAR(NOW()) - YEAR(data_nascimento);
RETURN idade;
END;
```

Com essa função, você pode calcular a idade de uma pessoa facilmente em suas consultas.

Função de tabela

Essas aceitam parâmetros e retornam uma tabela como resultado. Elas são úteis para encapsular consultas complexas e reutilizáveis. Por exemplo, suponha que você tenha uma função chamada OBT_TODOS_OS_PRODUTOS que aceita um parâmetro de categoria e retorna todos os produtos dessa categoria. Você pode usá-la em suas consultas para obter uma lista de produtos com base na categoria desejada.

```
CREATE FUNCTION OBT_TODOS_OS_PRODUTOS (categoria VARCHAR(50))
RETURNS TABLE
BEGIN
```

```

    RETURN (
        SELECT * FROM produtos WHERE categoria = categoria
    );
END;

```

Com essas funções criadas pelo usuário, você pode estender as capacidades do SQL e tornar suas consultas mais poderosas e flexíveis.

Agora que sabemos como as funções do SQL podem nos ajudar a realizar cálculos e manipulações de dados de forma eficiente, é hora de falar sobre os procedimentos, outra ferramenta poderosa. As funções são mais como scripts completos que podem fazer várias coisas em um banco de dados, mas são apenas pequenos blocos de código que retornam um valor específico.

Vamos dar um passo adiante em nossa jornada de aprendizado e descobrir como os procedimentos podem simplificar tarefas complexas e repetitivas em bancos de dados.

PROCEDIMENTOS

Com procedimentos, você pode agrupar várias instruções SQL em um único bloco lógico, que pode ser chamado e executado várias vezes conforme necessário. Isso é útil para automatizar processos, reduzir a redundância de código e melhorar a consistência dos dados.

Imagine que você tenha uma série de operações que precisa executar regularmente em seu banco de dados, como atualizar o status de um pedido, enviar notificações por e-mail aos clientes ou gerar relatórios mensais. Em vez de escrever manualmente cada instrução SQL toda vez que precisar realizar essas tarefas, você pode criar um procedimento que encapsula todas essas operações em um único bloco de código.

Por exemplo, você pode criar um procedimento chamado “AtualizarStatusPedido”, que aceita o ID do pedido como parâmetro e atualiza o status do pedido para “Processado”. Em seguida, você pode simplesmente chamar esse procedimento sempre que precisar atualizar o status de um pedido, em vez de escrever novamente todas as instruções SQL.

```

CREATE PROCEDURE inserir_produto (nome_produto VARCHAR(100), preco
DECIMAL (10, 2))

BEGIN
    INSERT INTO produtos (nome, preco) VALUES (nome_produto, preco);
END

```

Este procedimento recebe dois parâmetros: o nome do produto e o preço. Ele então insere um novo registro na tabela “produtos” com esses valores.

Agora, sempre que você quiser adicionar um novo produto, basta chamar este procedimento e passar o nome e o preço como argumentos:

```
CALL inserir_produto('Camiseta', 29.99);
```

Isso adicionará uma nova entrada na tabela “produtos” com o nome “Camiseta” e o preço de 29.99. Os procedimentos são úteis para automatizar tarefas repetitivas e complexas em bancos de dados SQL.

ATENÇÃO

Os procedimentos também podem aceitar parâmetros de entrada e retornar valores de saída, tornando-os ainda mais flexíveis e poderosos. Isso permite que você crie procedimentos genéricos que podem ser reutilizados em diferentes partes do seu sistema, economizando tempo e esforço de desenvolvimento.

Em resumo, os procedimentos são uma ferramenta valiosa para simplificar o desenvolvimento de aplicativos, melhorar a eficiência do código SQL e aumentar a produtividade dos desenvolvedores de banco de dados.