

Módulo 9: Funções e Procedimentos Armazenados

Módulo 9: Funções e Procedimentos Armazenados

No MySQL, funções e procedimentos armazenados são ferramentas que ajudam a realizar tarefas complexas e repetitivas de forma eficiente, promovendo a reutilização de código e reduzindo a necessidade de scripts extensos.

1. Funções no MySQL

Funções no MySQL são usadas para realizar operações específicas, como manipulação de strings, cálculos matemáticos e manipulação de datas e horas.

Funções de Manipulação de Strings

1. **CONCAT**: Concatena (junta) strings.

```
SELECT CONCAT('Olá, ', 'Mundo!') AS saudacao;  
-- Resultado: Olá, Mundo!
```

2. **SUBSTRING**: Extrai parte de uma string.

```
SELECT SUBSTRING('MySQL é poderoso', 7, 2) AS extrato;  
-- Resultado: é
```

3. **LOWER** e **UPPER**: Converte strings para letras minúsculas ou maiúsculas.

```
SELECT LOWER('MySQL') AS minusculas, UPPER('mysql') AS maiusculas;  
-- Resultado: minusculas: mysql | maiusculas: MYSQL
```

Aplicabilidade

- Gerar identificadores únicos:

```
SELECT CONCAT(id_cliente, '-', LOWER(nome)) AS identificador FROM clientes;
```

Funções de Data e Hora

1. **NOW**: Retorna a data e hora atuais.

```
SELECT NOW() AS data_hora_atual;
```

2. **CURDATE**: Retorna a data atual.

```
SELECT CURDATE() AS data_atual;
```

3. **DATE_FORMAT**: Formata uma data para um formato específico.

```
SELECT DATE_FORMAT(NOW(), '%d/%m/%Y %H:%i:%s') AS data_formatada;  
-- Resultado: 31/12/2024 14:30:45
```

4. **DATEDIFF**: Calcula a diferença em dias entre duas datas.

```
SELECT DATEDIFF('2024-12-31', '2024-12-25') AS diferenca_dias;  
-- Resultado: 6
```

Aplicabilidade

- Relatórios com intervalos de datas:

```
SELECT id_pedido, DATEDIFF(CURDATE(), data_pedido) AS dias_atraso  
FROM pedidos  
WHERE status = 'atrasado';
```

Funções Matemáticas

1. **ROUND**: Arredonda um número para o número de casas decimais especificado.

```
SELECT ROUND(123.456, 2) AS arredondado;  
-- Resultado: 123.46
```

2. **FLOOR** e **CEIL**: Arredondam para baixo ou para cima, respectivamente.

```
SELECT FLOOR(123.456) AS para_baixo, CEIL(123.456) AS para_cima;  
-- Resultado: para_baixo: 123 | para_cima: 124
```

3. **ABS**: Retorna o valor absoluto.

```
SELECT ABS(-50) AS absoluto;  
-- Resultado: 50
```

Aplicabilidade

- Cálculo de totais e arredondamentos para relatórios financeiros.

2. Procedimentos Armazenados (Stored Procedures)

O que são Procedimentos Armazenados

Um **procedimento armazenado** é um conjunto de instruções SQL que pode ser executado como uma unidade, permitindo a execução de tarefas complexas de forma repetitiva e eficiente.

- **Vantagens:**
 - Redução de redundância de código.
 - Melhoria na performance, pois o código é armazenado no servidor.
 - Maior segurança com controle de acesso.

Criação de Procedimentos

Sintaxe Geral

```
CREATE PROCEDURE nome_procedimento(parametros)  
BEGIN  
    corpo_do_procedimento;  
END;
```

Exemplo Prático

Criar um procedimento que lista todos os pedidos de um cliente específico:

```
CREATE PROCEDURE listar_pedidos_cliente(IN cliente_id INT)
BEGIN
    SELECT * FROM pedidos WHERE id_cliente = cliente_id;
END;
```

- **IN:** Parâmetro de entrada.

Execução de Procedimentos Armazenados

Os procedimentos armazenados são executados com o comando **CALL**.

Exemplo: Executar o procedimento criado:

```
CALL listar_pedidos_cliente(1);
```

3. Funções Definidas pelo Usuário (UDFs)

O que são UDFs

As **Funções Definidas pelo Usuário (User-Defined Functions)** permitem criar funções customizadas no MySQL para atender a necessidades específicas.

Criação de Funções Customizadas

Sintaxe Geral

```
CREATE FUNCTION nome_funcao(parametros)
RETURNS tipo_de_retorno
DETERMINISTIC
BEGIN
    corpo_da_funcao;
    RETURN valor;
END;
```

Exemplo Prático

Criar uma função que calcula o desconto aplicado a um valor:

```
CREATE FUNCTION calcular_desconto(valor DECIMAL(10,2), desconto DECIMAL(10,2))  
RETURNS DECIMAL(10,2)  
DETERMINISTIC  
BEGIN  
    RETURN valor - (valor * desconto / 100);  
END;
```

Executar a função:

```
SELECT calcular_desconto(200, 10) AS valor_com_desconto;  
-- Resultado: 180.00
```

Aplicabilidade

1. Funções no MySQL:

- Manipulação de strings em relatórios, como transformar nomes em formatos específicos.
- Cálculos matemáticos para criar análises financeiras detalhadas.
- Operações com datas, como calcular vencimentos e prazos.

2. Procedimentos Armazenados:

- Tarefas rotineiras, como gerar relatórios de vendas ou atualizar estoques.
- Processos complexos, como validações e cálculos automáticos durante a inserção de dados.

3. Funções Customizadas:

- Criação de cálculos específicos reutilizáveis em consultas, como margens de lucro ou taxas personalizadas.

Com o uso de funções e procedimentos armazenados, é possível criar sistemas mais eficientes, organizados e fáceis de manter. Isso melhora a performance e facilita a implementação de regras de negócio diretamente no banco de dados.

Vamos Praticar

Parte 1: Questões Práticas

- Manipulação de Strings:**
 - Crie uma consulta que utilize a função **CONCAT** para gerar um identificador único combinando o nome e o ID de um cliente.
 - Função **SUBSTRING**:**
 - Extraia os três primeiros caracteres do nome de um produto na tabela **produtos**.
 - Uso de **LOWER** e **UPPER**:**
 - Converta todos os nomes de clientes para letras maiúsculas em uma consulta SQL.
 - Funções de Data e Hora:**
 - Escreva uma consulta que utilize **NOW** para exibir a data e hora atuais, formatando a saída para o formato **DD/MM/AAAA HH:MM:SS** com **DATE_FORMAT**.
 - Diferença de Datas:**
 - Calcule o número de dias entre a data de hoje e a data de um pedido na tabela **pedidos** usando a função **DATEDIFF**.
 - Arredondamento:**
 - Arredonde o valor de um campo de preço para duas casas decimais usando a função **ROUND**.
 - Arredondar para Cima/Para Baixo:**
 - Use **FLOOR** para arredondar para baixo e **CEIL** para arredondar para cima o valor de um campo de total de vendas.
 - Criação de Procedimentos Armazenados:**
 - Crie um procedimento armazenado que insira um novo pedido na tabela **pedidos**, recebendo como entrada o ID do cliente e a data do pedido.
 - Execução de Procedimentos Armazenados:**
 - Execute o procedimento criado no exercício anterior para adicionar um pedido com seus parâmetros.
 - Criação de Funções Definidas pelo Usuário (UDFs):**
 - Crie uma função que calcule o valor total de um pedido aplicando um desconto percentual e use-a em uma consulta.
-

Parte 2: Questões Teóricas

- Definição de Funções:**
 - O que são funções no MySQL e quais os tipos mais comuns?
- Função **CONCAT**:**
 - Explique como a função **CONCAT** pode ser usada para manipular dados em relatórios.
- Funções de Data e Hora:**

- Qual a diferença entre **NOW** e **CURDATE**? Cite exemplos de suas aplicações.
- 4. **Arredondamento:**
 - Qual é a diferença entre as funções **ROUND**, **FLOOR** e **CEIL**? Em que situações cada uma é mais adequada?
- 5. **Vantagens dos Procedimentos Armazenados:**
 - Liste três vantagens de usar procedimentos armazenados em um banco de dados MySQL.
- 6. **Parâmetros de Procedimentos:**
 - Qual a diferença entre os tipos de parâmetros **IN**, **OUT** e **INOUT** em procedimentos armazenados?
- 7. **Execução de Procedimentos:**
 - Qual comando é usado para executar um procedimento armazenado e como ele funciona?
- 8. **Funções Definidas pelo Usuário (UDFs):**
 - O que são UDFs e como elas diferem de procedimentos armazenados?
- 9. **Aplicações de Funções Personalizadas:**
 - Explique uma situação prática onde a criação de uma função definida pelo usuário seria mais vantajosa do que o uso de uma função padrão.
- 10. **Funções e Procedimentos:**
 - Compare as funções e os procedimentos armazenados em termos de retorno de valores e uso em consultas.

Instruções para a Entrega das Atividades

1. **Elaboração e Envio do Arquivo**
 - Responda todas as questões de forma clara e objetiva.
 - Gere um arquivo no formato **.PDF** contendo as respostas de cada questão.
 - Envie o arquivo para os e-mails dos professores responsáveis.
2. **Validação da Atividade**
 - Após o envio do arquivo, procure o(s) professor(es) para realizar a validação da atividade.
 - **Não inicie a próxima atividade sem antes validar a anterior com o professor.**
3. **Forma de Validação**
 - **Explicação Verbal:** Explique cada resposta verbalmente ao(s) professor(es).
 - **Perguntas e Respostas:** Esteja preparado para responder aos questionamentos do(s) professor(es) sobre o conteúdo das respostas.
 - **Orientação:** Receba orientações sobre a apresentação do(s) tema(s).