

O que são funções

funções são blocos de comandos que realizam operações específicas, podendo receber parâmetros de entrada, processá-los e retornar um valor.

Diferença entre funções e procedimentos: Funções retornam um valor único, enquanto procedimentos podem retornar múltiplos valores e realizar tarefas mais complexas.

Por que Usar Funções?

Benefícios: Discutir como as funções podem simplificar consultas SQL complexas, promover a reutilização de código e melhorar a manutenção do banco de dados.

Tipos de Funções no MySQL

Funções Nativas

Funções nativas do MySQL para operações matemáticas, manipulação de strings, tratamento de datas, entre outras.

Exemplos: CONCAT(), DATE(), SUM(), AVG().

Criação de Funções

Sintaxe Básica para Criação de Funções

Sintaxe para a criação de funções no MySQL, destacando elementos como o nome da função, parâmetros, tipo de retorno e o corpo da função.

```
CREATE FUNCTION nome_da_funcao (parametro1 tipo1, parametro2 tipo2, ...)
```

```
RETURNS tipo_de_retorno
```

```
[characteristics...]
```

```
BEGIN
```

```
-- Corpo da função
```

```
DECLARE variavel1 tipo1;
```

```
...
```

```
SET variavel1 = valor1;
```

```
...
```

```
RETURN (valor_de_retorno);
```

```
END;
```

Elementos da Sintaxe

CREATE FUNCTION: Este comando inicia a definição de uma nova função.

nome_da_funcao: É o nome atribuído à função. Este nome será usado para chamar a função em outras consultas SQL.

parâmetros: As funções podem receber zero ou mais parâmetros. Cada parâmetro é definido pelo seu nome seguido pelo tipo de dado. Os parâmetros funcionam como variáveis de entrada que a função pode utilizar em suas operações internas.

RETURNS tipo_de_retorno: Define o tipo do valor que será retornado pela função. Todo procedimento no MySQL deve retornar um valor, e esse valor pode ser de qualquer tipo de dado suportado pelo MySQL, como INT, VARCHAR, DATE, etc.

BEGIN...END: Delimita o corpo da função, onde a lógica da função é implementada. Dentro deste bloco, você pode declarar variáveis locais, controlar estruturas como loops e condições, e utilizar comandos SQL.

DECLARE: Dentro do corpo da função, você pode declarar variáveis locais que serão usadas para armazenar dados temporariamente.

SET: Utilizado para atribuir valores às variáveis declaradas dentro da função.

RETURN: Este comando especifica o valor a ser retornado pela função. O tipo deste valor deve corresponder ao tipo definido no RETURNS.

Boas Práticas na Criação de Funções

Enfatizar a importância de nomes significativos para funções e parâmetros.

Discutir a relevância de comentar o código e documentar o propósito e uso das funções.

Exemplos Práticos

Função para Formatar Telefone

Com base nos campos TELEFONE presentes em várias tabelas, essa função pode formatar números de telefone para um padrão uniforme.

```
DELIMITER $$
```

```
CREATE FUNCTION `formatar_telefone` (telefone VARCHAR(10))
```

```
RETURNS VARCHAR(14)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    RETURN CONCAT('(', SUBSTRING(telefone, 1, 2), ') ', SUBSTRING(telefone, 3, 4), '-',  
SUBSTRING(telefone, 7, 4));
```

```
END$$
```

```
DELIMITER ;
```