

PROCEDIMENTOS ARMAZENADOS

(STORED PROCEDURES)

Continuação

Estruturas de Programação em Procedimentos

- ❑ Os **procedimentos armazenados** são estruturas extremamente poderosas que facilitam a interação dos usuários (Aplicação) com o servidor de banco de dados (SGBD), além de ajudar no controle de acesso aos dados, aumentando a segurança do sistema;
- ❑ Um dos requisitos que contribuem para essas características dos procedimentos está na possibilidade de uso de estruturas de controle de fluxo de dados, como estruturas de **decisão** e de **repetição**.
- ❑ Você está familiarizado com algumas dessas estruturas, pois já foram apresentadas nas disciplinas de lógica e linguagem de programação;

Estruturas de Programação em Procedimentos

- Vamos estudar as duas mais usadas, o **IF**, que define estruturas de decisão, e o **WHILE**, que define estruturas de repetição;
- O destaque abaixo apresenta a estrutura do **IF** em **MySQL**.
- **IF (condição 1) AND (condição 2) THEN**
 - comandos em SQL
- **ELSE**
 - comandos em SQL
- **END IF;**

IF em Procedimentos

- A sintaxe pode ser explicada do seguinte modo:
- **SE (IF)** a **condição 1** e a **condição 2** forem **satisfeitas**, um grupo de comandos em SQL é executado, **SENÃO (ELSE)** o último grupo de comando é executado;
- Podemos utilizar **UMA** ou **VÁRIAS** condições dentro de um IF, sempre separadas por **operadores lógicos (OR ou AND)**;
- Também é possível criar um IF dentro de outro IF, fazendo várias condições diferentes. Veja o código a seguir:

Sintaxe com 02 IFs

□ IF (**condição 1**) AND (**condição 2**) THEN

□ IF (**condição 3**) THEN

■ comandos em SQL

□ ELSE

■ comandos em SQL

□ END IF;

Estrutura do 2º SE (IF)

□ ELSE

■ comandos em SQL

□ END IF;

Estrutura do 1º SE (IF)

ATENÇÃO: Observe que a **IDENTAÇÃO** (recuo) é **muito** importante para identificar onde começa e termina um IF!

IF em Procedimentos

- Para exemplificar a utilização da estrutura IF em um **procedimento armazenado**, imagine que precisamos criar um procedimento para inserir clientes no nosso **banco de dados do Hotel**, mas, para isso, devemos **garantir** que sejam informados o **nome** e o **cpf** do cliente.
- O procedimento a seguir apresenta uma solução para esse problema. Analise com atenção a sequência de comandos apresentada.

Exemplo de Procedimento com IF

- ❑ CREATE PROCEDURE **InserirCliente** (**nome** varchar(100), **cpf** varchar(20), **telefone** varchar(50), **celular** varchar(50), **endereco** int)
- ❑ BEGIN
- ❑ IF ((**nome** <> '') AND (**cpf** <> '')) THEN
 - ❑ INSERT INTO cliente (nome_cli, cpf_cli, telefone_cli, celular_cli, cod_end) VALUES (**nome**, **cpf**, **telefone**, **celular**, **endereco**);
 - ❑ SELECT 'Cliente inserido com sucesso!' as Confirmação;
- ❑ ELSE
 - ❑ SELECT 'NOME e CPF devem ser fornecidos para o cadastro!' as Erro;
- ❑ END IF;
- ❑ END;

Análise de Procedimento com IF

- Observe que **SE** os respectivos parâmetros (variáveis) **não forem vazios** (“”) o comando *insert* será executado e uma mensagem de confirmação será exibida;
- **SENÃO** o *insert* não irá ser executado e será apresentada na tela uma mensagem de erro;
- A condição do IF é definida usando o operador “<>” que significa DIFERENTE DE e o operador lógico AND, ou seja, SE nome DIFERENTE DE **vazio** E cpf DIFERENTE DE **vazio**, a operação de inserção é executada.
- **ATENÇÃO:** Para cada **IF** criado deve existir um **END IF**;

While em Procedimentos

- Agora que vimos a utilização das estruturas de decisão, vamos discutir um dos exemplos de estrutura de repetição. O código a seguir apresenta a estrutura do WHILE em *MySQL*.
- A sintaxe pode ser explicada do seguinte modo: **ENQUANTO (WHILE)** a condição é satisfeita **EXECUTE (DO)** um grupo de comandos em SQL.
- **WHILE** condição **DO**
- comandos em SQL
- **END WHILE**

Exemplo de Procedimento com While

- Para exemplificar a utilização da estrutura WHILE ... END WHILE num **procedimento armazenado**, imagine que:
 1. Precisamos criar um procedimento para reajustar os preços da tabela **produto** do nosso **banco de dados de Hotel**.
 - a) O usuário deve informar qual é o **percentual** a ser aumentado;
 - b) O usuário deve dizer qual o **valor máximo** que o produto pode chegar;
 - c) Após cada reajuste o **valor atualizado** do produto deve ser mostrado para o usuário;
- O procedimento a seguir apresenta uma solução para esse problema. Veja:

Exemplo de Procedimento com While

- ❑ CREATE PROCEDURE **aumentarProduto** (**codigo** int, **preco** double, **percentual** int)
- ❑ **BEGIN**
- ❑ **WHILE** ((*SELECT valor_prod FROM produto WHERE cod_prod = **codigo***) < **preco**) **DO**
 - ❑ **UPDATE** produto **SET** valor_prod = valor_prod + (valor_prod * (**percentual**/100)) **WHERE** cod_prod = **codigo**;
 - ❑ **SELECT** **CONCAT**('Produto reajustado para: ', (*SELECT valor_prod FROM produto WHERE cod_prod = **codigo***)) **AS** Mensagem;
- ❑ **END WHILE**;
- ❑ **END**;

- ❑ **call** aumentarProduto(1, 20, 5);

Análise do Comando While

- ❑ Analisando o procedimento anterior, verifica-se que o usuário deseja atualizar o valor do produto de código nº 01 em 5 % por cento até que atinja o valor máximo de R\$ 20.00;
- ❑ Assim, o procedimento irá atualizar o valor do produto aplicando um aumento através da porcentagem inserida pelo usuário, mostrando o valor do aumento em cada “passada” do While;
- ❑ Observe que na mensagem de retorno, foi usado a função **CONCAT** para **unir as palavras**;

Declaração de Variáveis

- É Possível declarar variáveis dentro dos procedimentos. Para isso utilize da sintaxe: **DECLARE** nome_da_variável tipo_variável;
- Também é possível atribuir valores para essa variável.
- A primeira forma é através do **INTO** pegando o valor de um **ATRIBUTO da TABELA** e jogando dentro da **VARIÁVEL**;
- A segunda forma é atribuir **diretamente** o valor usando a sintaxe: **SET** nome_da_variável = valor_desejado;

Uso de Variáveis com INTO

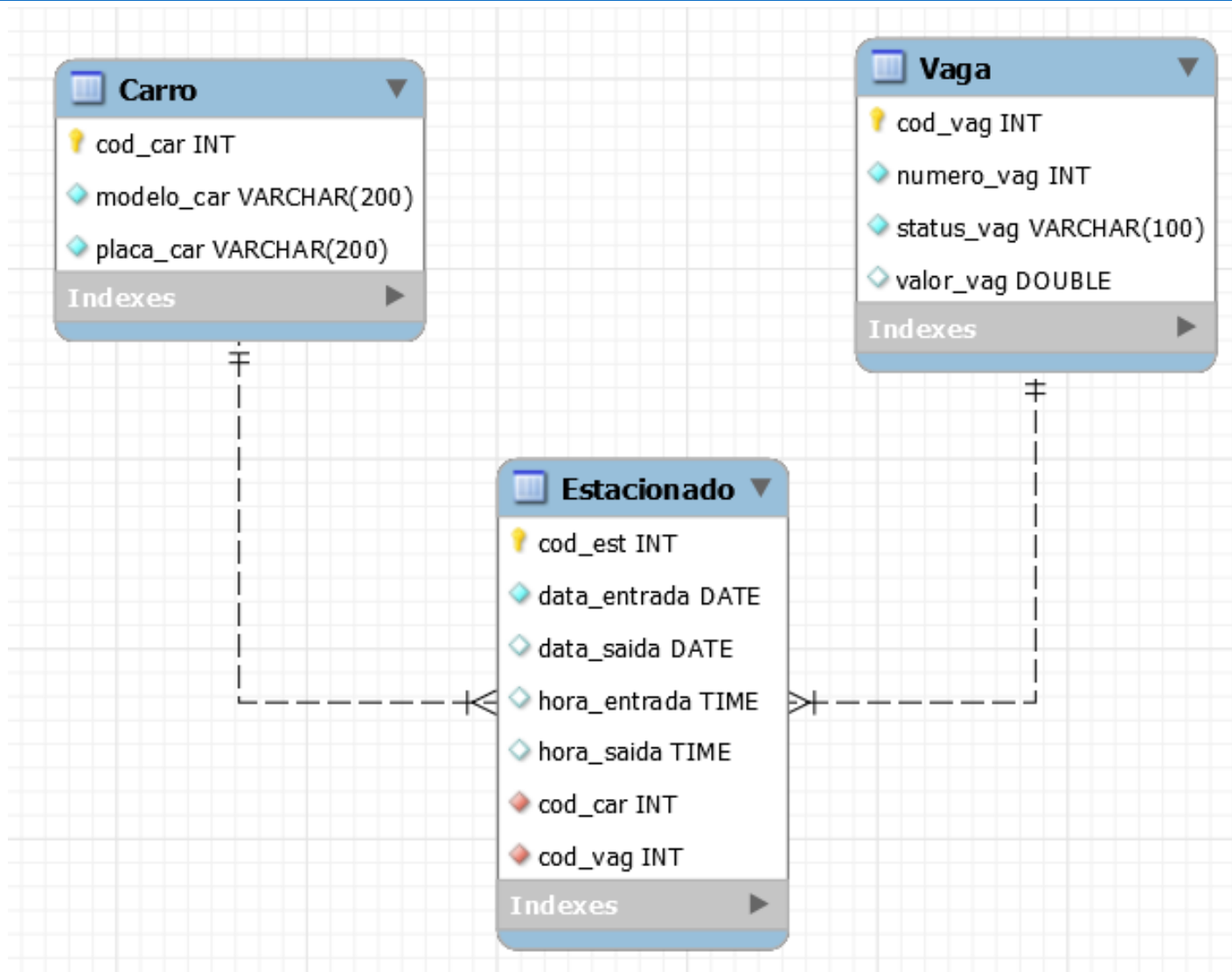
- ❑ CREATE PROCEDURE tipoCliente (**cli** int)
- ❑ BEGIN
- ❑ **DECLARE** **soma** FLOAT;
- ❑ SELECT SUM(valorTotal_hosp) **INTO** **soma** FROM hospedagem WHERE cod_cli = **cli**;
- ❑ IF (soma >= 200) THEN
 - ❑ SELECT 'O Cliente é VIP!' as Tipo;
- ❑ ELSE
 - ❑ SELECT 'O Cliente é Simples!' as Tipo;
- ❑ END IF;
- ❑ END;

Uso de Variáveis com SET

- ❑ CREATE PROCEDURE somarNumeros (n1 float, n2 float)
- ❑ BEGIN
- ❑ DECLARE resultado FLOAT;
- ❑ SET resultado = n1 + n2;
- ❑ SELECT CONCAT('O resultado da soma é ', resultado) as Resultado;
- ❑ END;

Exemplos de Procedimentos com IF e ELSE

Diagrama de Referência



Exemplo 1 – Comparando com 01 condição

```
DELIMITER $$  
CREATE PROCEDURE selecionarCarro (carro Int)  
BEGIN  
    IF (carro IS NOT NULL) THEN  
        SELECT * FROM CARRO WHERE COD_CAR = carro;  
    ELSE  
        SELECT 'Digite o Código de um carro válido!' as Erro;  
    END IF;  
    END  
$$ DELIMITER ;
```

Exemplo 2 – Comparando com 02 condições

- ❑ DELIMITER \$\$
- ❑ CREATE PROCEDURE **selecionar_carro_ComLike** (**modelo** varchar(50))
- ❑ BEGIN
- ❑ IF ((**modelo** IS NOT NULL) AND (**modelo** <> '')) THEN
 - ❑ SELECT * FROM CARRO WHERE MODELO_CAR **LIKE CONCAT**(**modelo**, '%');
- ❑ ELSE
 - ❑ SELECT 'Digite o código de um carro válido!' as Erro;
- ❑ END IF;
- ❑ END
- ❑ \$\$ DELIMITER ;

Exemplo 3 – Comparando com 02 IF

- ❑ DELIMITER \$\$
- ❑ CREATE PROCEDURE **selecionar_carro_ComLike** (**modelo** varchar(50))
- ❑ BEGIN
- ❑ IF ((**modelo** IS NOT NULL) THEN
 - ❑ IF (**modelo** <> '') THEN
 - SELECT * FROM CARRO WHERE MODELO_CAR **LIKE CONCAT**(**modelo**, '%');
 - ❑ ELSE
 - Select 'O modelo digitado está com texto VAZIO!' as Erro;
 - ❑ END IF;
- ❑ ELSE
 - Select 'O modelo digitado está com texto NULO!' as Erro;
- ❑ END IF;
- ❑ END;
- ❑ \$\$ DELIMITER ;

EXERCÍCIOS

BASE: BD_HOTEL

Exercício I

- ❑ Crie um procedimento que se comporte como uma **calculadora**. Você deverá receber **02 números** como parâmetros de **entrada**. Também deverá receber um parâmetro de entrada que informe o tipo da operação (+, -, /, *);
- ❑ Faça estruturas IF para calcular os 02 números de acordo com a operação (símbolo) informado pelo usuário;
- ❑ Retorne o resultado em uma mensagem de retorno;

Exercício II

- Crie um procedimento para informar a classificação de cada cliente utilizando as seguintes **verificações** (ifs):
 - Caso a soma das hospedagens realizadas pelo cliente seja maior que R\$ 1000 sua categoria é cliente VIP;
 - Caso seja entre R\$ 500 e 1000 sua categoria é cliente NORMAL;
 - caso seja menor que R\$ 500 sua categoria é POPULAR;
- O código do cliente deve ser passado como parâmetro de entrada.
- Na mensagem de retorno deve ser informado o **nome do cliente** e o seus **status** na classificação.

Exercício III

- Crie um procedimento que reajuste o preço dos produtos em estoque. Você deve receber como parâmetro de entrada a porcentagem a ser reajustada e uma variável que informe se o reajuste é para **aumentar** ou **diminuir** o preço. Faça as seguintes verificações:
 - ▣ Se o reajuste for para **aumentar** não permita que seja maior do 50%;
 - ▣ Se o reajuste for para **diminuir** não permita que seja maior do 20%;
- Retorno mensagens de retorno cada a atualização aconteça ou não de acordo com a condição;