

# BANCO DE DADOS I



## STORED PROCEDURE

Prof: Ricardo Rufino

# Stored Procedure



- São funções escritas usando SQL e ficam armazenadas no servidor
- Podem receber parâmetros e executar funções complexas, retornando ou não as informações para o usuário
  - Vantagens:
    - Minimizam o tráfego da rede
    - São mais rápidos, aproveitam a capacidade do servidor e da otimização do SGBD
    - Facilitam a manutenção. Pode ser acessada por diversos programas, se houver alguma modificação a ser feita, basta fazê-la no BD
  - Limitações:
    - A linguagem de escrita de stored procedures e triggers pode ser muito limitada para operações mais complexas.

# Sintaxe



```
CREATE PROCEDURE <nome> (<param1>[tipo], <param2>[tipo],...)
```

```
BEGIN
```

```
    DECLARE <variavel>[tipo];
```

```
        <corpo da procedure>
```

```
END;
```

- Os parâmetros podem ser:
  - IN → Entrada
  - OUT → Saída ou retorno
  - INOUT → Entrada e saída
- Para invocar uma procedure basta invocar **CALL** <nome>(<param1>, <param2>, ...)

# Exemplo 01 - Olá

```
DROP PROCEDURE IF EXISTS exemplo01;
```

```
CREATE PROCEDURE exemplo01()
```

```
BEGIN
```

```
    SELECT 'Olá Mundo !!!';
```

```
END;
```

```
CALL exemplo01();
```

# Exemplo 02 - Insert

```
CREATE PROCEDURE insere_produto(nome  
varchar(60))
```

```
BEGIN
```

```
    IF(nome<>"")
```

```
    THEN
```

```
        INSERT INTO produto (prod_nome) values(nome);
```

```
    ELSE
```

```
        SELECT 'A descrição deve ser fornecida para o cadastro';
```

```
    END IF;
```

```
END;
```

```
CALL insere_produto('Skol 600ml');
```

# Exemplo 03 - Retorno

```
CREATE PROCEDURE insere_produto(nome varchar(60),  
OUT ultimoInserido INT)
```

```
BEGIN
```

```
    IF(nome<>'')
```

```
    THEN
```

```
        INSERT INTO produto (prod_nome) values(nome);
```

```
        SELECT MAX(prod_id) into ultimoInserido FROM produto;
```

```
    ELSE
```

```
        SELECT 'A descrição deve ser fornecida para o cadastro';
```

```
    END IF;
```

```
END;
```

```
CALL insere_produto('Skol 1L', @ultimoInserido);
```

```
SELECT @ultimoInserido;
```

# Exemplo 04 - Exclusão

```
CREATE PROCEDURE  
exclui_produto(id_produto int)
```

```
BEGIN
```

```
    DELETE FROM produto WHERE prod_id=id_produto;
```

```
END;
```

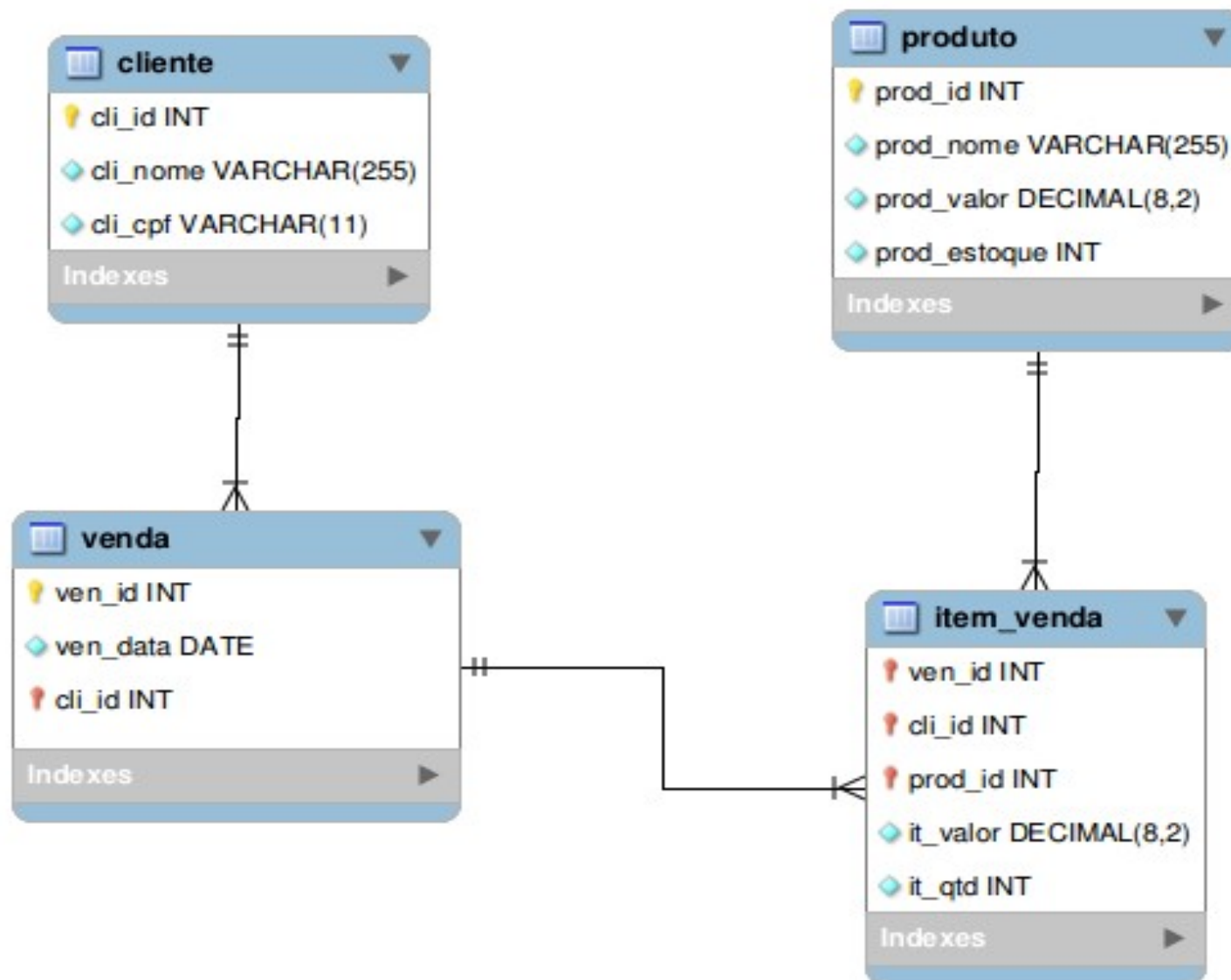
```
CALL exclui_produto(1);
```

# Objetivos

- Desenvolver um processo de vendas:
  - Efetuar vendas através de **procedures**
  - Movimentar estoque através de **triggers**
  - Retornar média de venda por cliente através de **functions**
  - Criar **views** para consulta da dados referente as vendas



# DER



# Observações

- Abra o gerenciador do MySQL:
  - Execute o arquivo `script_banco.sql` para criar as tabelas do banco de dados.
  - Posteriormente execute o arquivo `inserts.sql`
- Resolva os 3 exercícios contidos no arquivo `Exercícios_procedure.pdf`.