

## Lista 03 – Laboratório SQL – CAP 349 – Bancos de Dados Geográficos

Considere o diagrama relacional da Figura 1 e faça os exercícios abaixo. No final do laboratório, cada aluno deve entregar um script com todos os comandos SQL executados nos exercícios.

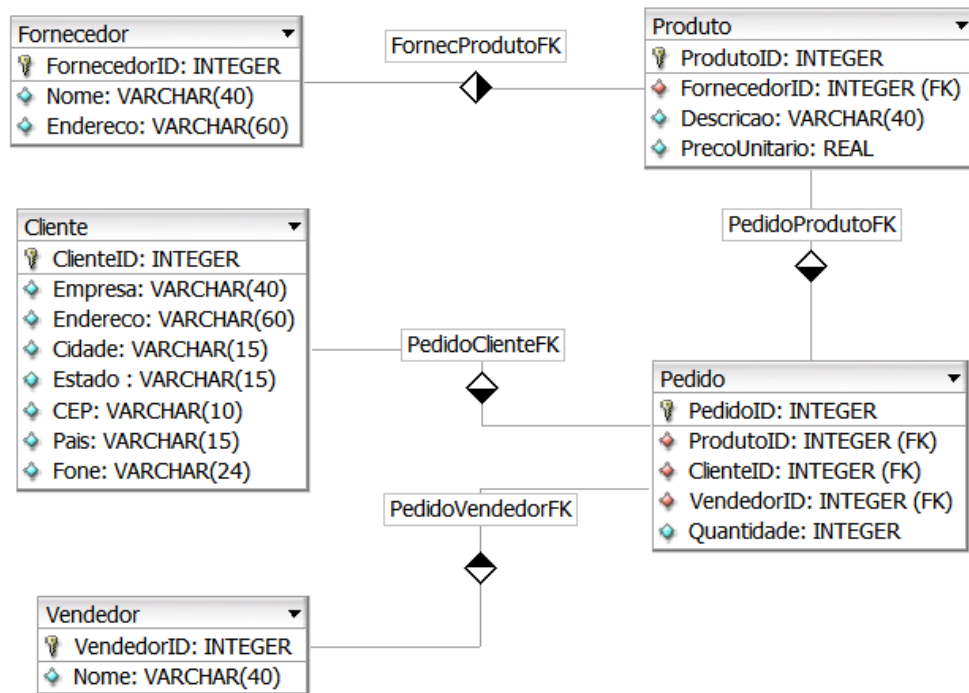


Figure 1 – Diagrama relacional.

Exercícios:

- 01) Crie as tabelas Fornecedor e Produto.
- 02) Use os comandos do arquivo “SQL\_scripts.sql” para criar as tabelas Cliente, Vendedor e Pedido.
- 03) Insira todos os registros do arquivo “SQL\_scripts.sql”. O SGBD retornará um erro ao inserir os registros da tabela Cliente. Qual o comando SQL é utilizado para solucionar esse erro e inserir todos os registros da tabela Cliente?
- 04) Recupere as informações do catálogo:
  - a. Quais esquemas existem nesse banco de dados?
  - b. Recupere as informações sobre as tabelas do esquema “public”.

- c. Recupere as informações sobre todas as colunas de todas as tabelas do esquema "public".
  - d. Recupere as informações sobre todas as restrições (*constraints*) de todas as tabelas do esquema "public".
- 05) Selecione todos os vendedores ordenados pelo nome.
  - 06) Quantos vendedores estão cadastrados no banco de dados?
  - 07) Quantos pedidos foram feitos pelo vendedor 'Jose Marcio'?
  - 08) Quantos itens foram vendidos pelo vendedor 'Jose Marcio'?
  - 09) Quantos itens foram vendidos por cada vendedor?
  - 10) Quais vendedores venderam mais que 300 itens?
  - 11) Quanto foi o valor de cada pedido?
  - 12) Quanto cada vendedor vendeu?
  - 13) Quanto foi a comissão de cada vendedor (considerando que a comissão é 10% do valor da venda)
  - 14) Crie uma view que contenha as seguintes informações sobre os pedidos: identificador do pedido, nome do vendedor, nome do cliente, endereço do cliente, telefone do cliente, descrição do produto, fornecedor do produto e a quantidade do produto.
  - 15) Insira um novo pedido => vendedor: "Luis Claudio", cliente: "ACM", produto: "Tinta", quantidade: 112
  - 16) Altere o nome do vendedor "Andre Carlos" para "Andre Carlos Garcia"
  - 17) Reajuste o preço de todos os produtos em 10%
  - 18) Remova o cliente "ACM" da tabela cliente
- OBS: Observe o que acontece com os pedidos associados ao cliente "ACM"
- 19) Adicione um atributo "CNPJ" na tabela "Cliente" (os valores dos CNPJ devem ser únicos)
  - 20) Mude o atributo "Fone" da tabela "Cliente" para um tipo numérico
  - 21) Renomeie o atributo "CNPJ" para "CNPJ\_Cliente"
  - 22) Remova o atributo "CNPJ\_Cliente"
  - 23) Remova todos os registros da tabela "Vendedor"
- OBS: Observe o que acontece com os registros das tabelas que recebem o atributo "VendedorID" como foreign key

- 24) Remova o atributo "VendedorID" da tabela "Vendedor"
- 25) Remova a tabela "Vendedor"
- 26) Remova todas as tabelas do banco (o esquema e o conteúdo)
- 27) Crie novamente as tabelas do banco de dados, usando os comandos CREATE TABLE já utilizados.
- 28) Crie uma nova tabela chamada "vendedor\_comissao" para armazenar a comissão de cada vendedor. Essa tabela terá o id de cada vendedor, que será PRIMARY KEY e FOREIGN KEY, e o valor da comissão.
- 29) Use o recurso de trigger para popular automaticamente a tabela "vendedor\_comissao", conforme vendedores e pedidos são inseridos no banco de dados. Crie duas triggers:
  - a. Cada vez que um novo vendedor é inserido no banco de dados (tabela "vendedor"), a trigger deve inserir um novo registro na tabela "vendedor\_comissao" contendo o id desse novo vendedor e o valor 0.0 para sua comissão. Após criar essa trigger, execute os comandos INSERT TABLE da tabela "vendedor" e veja o que acontece:

```
CREATE FUNCTION insert_vendedor_comissao() RETURNS trigger
AS $vend_comissao$
    BEGIN
        INSERT INTO vendedor_comissao(vendedorid, comissao)
        VALUES (NEW.vendedorid, 0.0);
        RETURN NEW;
    END;
$vend_comissao$ LANGUAGE plpgsql;

CREATE TRIGGER vendedor_comissao_trigger_1
AFTER INSERT ON vendedor
FOR EACH ROW
EXECUTE PROCEDURE insert_vendedor_comissao();
```

- b. Cada vez que um novo pedido é inserido ou atualizado no banco de dados (tabela "pedido"), a trigger deve atualizar a comissão na tabela "vendedor\_comissao" do vendedor que efetuou o pedido.

Após criar essa trigger, execute os comandos INSERT TABLE da tabela “pedido” e veja o que acontece.

- 30) Insira uma nova coluna na tabela vendedor com o salário de cada uma:
- a. Jose Marcio: R\$ 3500.00
  - b. Luis Claudio: R\$ 3000.00
  - c. Andre Carlos: R\$ 2500.00
- 31) Altere a trigger acima para que nenhum vendedor ganhe uma comissão com o valor acima do seu salário. Ou seja, toda vez que o vendedor efetuar um pedido, a trigger deve checar se o novo valor da comissão é maior do que o salario do vendedor. Se for maior, o vendedor deve receber exatamente o valor do seu salário de comissão.