

## Respostas

- **Todas as respostas foram baseadas na sintaxe do PostgreSQL;**

1. Dados as três tabelas do esquema, é possível inferir que as chaves primárias são respectivamente:

- **tb\_customer: id\_customer;**
- **dm\_address\_type: cd\_address\_type;**
- **tb\_customer\_address: chave primária composta (id\_customer, cd\_address\_type), além disso o conjunto chave primária é composta por chaves estrangeiras das tabelas tb\_customer e dm\_address\_type respectivamente;**

2. Os comandos, em ordem, são respectivamente:

1. **INSERT INTO tb\_customer VALUES (nextval('tb\_customer\_id\_customer\_seq'), 888777666.55, 'Joãozinho Silva');**
2. **INSERT INTO tb\_customer\_address VALUES(currval('tb\_customer\_id\_customer\_seq'), 'R', 'Rua das Flores', 1, null, '01234-567');**
3. **INSERT INTO tb\_customer\_address VALUES(currval('tb\_customer\_id\_customer\_seq'), 'C', 'Rua das Pedras', 100, 'Conjunto 200', '01234-567');**

3. O número de endereços é dado em função da quantidade de tuplas válidas na tabela dm\_address\_type, no caso do esquema em questão temos três registros, R, C e O, que possibilitam um cliente possuir três endereços diferentes, se por acaso adicionarmos mais uma tupla na tabela dm\_address\_type, cada cliente terá a possibilidade de ter quatro endereços.

**Número de Endereços possíveis = Número de tuplas da Tabela dm\_address\_type**

4. Pensei em duas possibilidades:

I. **Duas funções SQL para deletar:**

(1) Primeiro deletamos na tabela que contém a chave estrangeira, no caso tb\_customer\_address, a tupla cujo o **tb\_customer\_address.id\_customer = tb\_customer.id\_customer(com cpf\_cnpj = 888777666.55).**

```
DELETE
FROM tb_customer_address
WHERE tb_customer_address.id_customer = ( SELECT id_customer
                                           FROM tb_customer
                                           WHERE cpf_cnpj = 888777666.55);
```

(2) Depois deletamos o valor na tabela tb\_customer, dado o **cpf\_cnpj= 888777666.55.**

```
DELETE
FROM tb_customer
WHERE cpf_cnpj = 888777666.55
```

## II. Com um Trigger(gatilho), automatizando a segunda exclusão:

(1) Defina uma função para deletar a tupla na tabela `tb_customer` comparando o `id_customer` da tupla que foi apagada na tabela `tb_customer_address`, e cujo `cpf = 888777666.55`, com o `id_customer` correspondente na tabela `tb_customer`. O gatilho para a chamada da função ocorre após o comando `DELETE` sobre a tabela `tb_customer_account`.

```
CREATE OR REPLACE FUNCTION delete_parent() RETURNS TRIGGER AS $$
BEGIN
    DELETE
    FROM tb_customer
    WHERE tb_customer.id_customer = OLD.id_customer;
    return NULL;
END;
$$
language plpgsql;
```

```
CREATE TRIGGER delete_on_cascade AFTER DELETE ON tb_customer_address
FOR EACH ROW
EXECUTE PROCEDURE delete_parent();
```

(2) Após definido **(1)**. É apenas necessário chamar a função de deletar na tabela que contém a chave estrangeira(filha), no caso tb\_customer\_address.

[illegible]