

# Functions da entidade cliente

```
CREATE OR REPLACE FUNCTION criar_cliente(  
    p_nome TEXT,  
    p_cpf TEXT,  
    p_data_nascimento DATE,  
    p_tipo_cliente TEXT,  
    p_rua TEXT,  
    p_numero TEXT,  
    p_complemento TEXT,  
    p_cidade TEXT,  
    p_estado TEXT,  
    p_cep TEXT  
) RETURNS BIGINT  
LANGUAGE plpgsql  
AS $$  
DECLARE  
    v_cliente_id BIGINT;  
BEGIN  
    INSERT INTO clientes (nome, cpf, data_nascimento, tipo_cliente)  
    VALUES (p_nome, p_cpf, p_data_nascimento, p_tipo_cliente)  
    RETURNING id INTO v_cliente_id;  
    INSERT INTO enderecos (cliente_id, rua, numero, complemento, cidade, estado,  
cep)  
    VALUES (v_cliente_id, p_rua, p_numero, p_complemento, p_cidade, p_estado,  
p_cep);  
    RETURN v_cliente_id;  
END;  
$$;
```

---

```
CREATE OR REPLACE FUNCTION atualizar_cliente(  
    p_id BIGINT,  
    p_nome TEXT,  
    p_data_nascimento DATE,
```

```

p_tipo_cliente TEXT,
p_rua TEXT,
p_numero TEXT,
p_complemento TEXT,
p_cidade TEXT,
p_estado TEXT,
p_cep TEXT
) RETURNS VOID
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE clientes
    SET nome = p_nome,
        data_nascimento = p_data_nascimento,
        tipo_cliente = p_tipo_cliente
    WHERE id = p_id;
    UPDATE enderecos
    SET rua = p_rua,
        numero = p_numero,
        complemento = p_complemento,
        cidade = p_cidade,
        estado = p_estado,
        cep = p_cep
    WHERE cliente_id = p_id;
END;
$$;

```

---

```

CREATE OR REPLACE FUNCTION listar_todos_clientes()
RETURNS TABLE (
    cliente_id BIGINT,
    nome VARCHAR(100),
    cpf VARCHAR(14),
    data_nascimento DATE,
    tipo_cliente VARCHAR(20),
    endereco_id BIGINT,
    endereco_cliente_id BIGINT,
    rua VARCHAR(100),
    numero VARCHAR(10),
    complemento VARCHAR(50),
    cidade VARCHAR(50),
    estado VARCHAR(2),
    cep VARCHAR(9)
)

```

```

) AS $$
BEGIN
    RETURN QUERY
    SELECT
        c.id AS cliente_id,
        c.nome,
        c.cpf,
        c.data_nascimento,
        c.tipo_cliente,
        e.id AS endereco_id,
        e.cliente_id AS endereco_cliente_id,
        e.rua,
        e.numero,
        e.complemento,
        e.cidade,
        e.estado,
        e.cep
    FROM clientes c
    LEFT JOIN enderecos e ON c.id = e.cliente_id;
END;
$$ LANGUAGE plpgsql;

```

---

```

CREATE OR REPLACE FUNCTION buscar_cliente_por_id(p_id BIGINT)
RETURNS TABLE (
    cliente_id BIGINT,
    nome VARCHAR(100),
    cpf VARCHAR(11),
    data_nascimento DATE,
    tipo_cliente VARCHAR(20),
    endereco_id BIGINT,
    rua VARCHAR(100),
    numero VARCHAR(10),
    complemento VARCHAR(50),
    cidade VARCHAR(100),
    estado VARCHAR(2),
    cep VARCHAR(9)
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY

```

**SELECT**

c.id,  
c.nome,  
c.cpf,  
c.data\_nascimento,  
c.tipo\_cliente,  
e.id,  
e.rua,  
e.numero,  
e.complemento,  
e.cidade,  
e.estado,  
e.cep

**FROM** clientes c

**JOIN** enderecos e **ON** e.cliente\_id = c.id

**WHERE** c.id = p\_id;

**END;**

**\$\$;**

---

**CREATE OR REPLACE FUNCTION** deletar\_cliente(p\_id **BIGINT**)

**RETURNS VOID**

**LANGUAGE** plpgsql

**AS \$\$**

**BEGIN**

**DELETE FROM** enderecos **WHERE** cliente\_id = p\_id;

**DELETE FROM** clientes **WHERE** id = p\_id;

**END;**

**\$\$;**