

Exercício 1: Trigger para Auditar Inserts em **cadastro_cliente**

1. Criação da Tabela de Logs

```
CREATE TABLE log_auditoria (  
    id SERIAL PRIMARY KEY,  
    cod_cliente INT,  
    nome_cliente VARCHAR(30),  
    sobrenome VARCHAR(30),  
    dt_cadastro DATE,  
    data_insert TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

2. Criação da Trigger

```
CREATE OR REPLACE FUNCTION trg_cad_clientes_func()  
RETURNS TRIGGER AS $$  
BEGIN  
    INSERT INTO log_auditoria (cod_cliente, nome_cliente, sobrenome,  
dt_cadastro)  
    VALUES (NEW.cod_cliente, NEW.nome_cliente, NEW.sobrenome,  
NEW.dt_cadastro);  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER trg_cad_clientes  
AFTER INSERT ON cadastro_cliente  
FOR EACH ROW  
EXECUTE FUNCTION trg_cad_clientes_func();
```

Exercício 2: Trigger para Armazenar Valores Antes de Excluir

1. Criação da Tabela de Logs para Exclusão

```
CREATE TABLE log_exclusao (  
    id SERIAL PRIMARY KEY,  
    cod_cliente INT,
```

```
    nome_cliente VARCHAR(30),
    sobrenome VARCHAR(30),
    dt_cadastro DATE,
    data_exclusao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Criação da Trigger

```
CREATE OR REPLACE FUNCTION trg_exclusao_cliente_func()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO log_exclusao (cod_cliente, nome_cliente, sobrenome,
dt_cadastro)
    VALUES (OLD.cod_cliente, OLD.nome_cliente, OLD.sobrenome,
OLD.dt_cadastro);
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_exclusao_cliente
BEFORE DELETE ON cadastro_cliente
FOR EACH ROW
EXECUTE FUNCTION trg_exclusao_cliente_func();
```

Exercício 3: Trigger para Auditar Aumentos de Salário

1. Criação da Tabela de Funcionários e Auditoria

```
CREATE TABLE salario (
    cd_func SERIAL PRIMARY KEY,
    nm_func VARCHAR(30),
    salario FLOAT
);
```

```
CREATE TABLE salario_aud (
    dt_audit TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    cd_func INT,
    nm_func VARCHAR(30),
```

```
    salario_ant FLOAT,  
    salario_novo FLOAT  
);
```

2. Criação da Trigger

```
CREATE OR REPLACE FUNCTION trg_func()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.salario - OLD.salario > 1500 THEN  
        INSERT INTO salario_aud (cd_func, nm_func, salario_ant,  
salario_novo)  
        VALUES (OLD.cd_func, OLD.nm_func, OLD.salario, NEW.salario);  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_auditoria_salario  
AFTER UPDATE ON salario  
FOR EACH ROW  
EXECUTE FUNCTION trg_func();
```

Exercício 4: Trigger para Auditar Troca de Departamento

1. Criação da Tabela de Logs para Troca de Departamento

```
CREATE TABLE log_troca_departamento (  
    id SERIAL PRIMARY KEY,  
    cd_funcionario INT,  
    nm_funcionario VARCHAR(30),  
    dept_ant VARCHAR(30),  
    dept_novo VARCHAR(30),  
    data_troca TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

2. Criação da Trigger

```

CREATE OR REPLACE FUNCTION trg_troca_departamento_func()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO log_troca_departamento (cd_funcionario,
nm_funcionario, dept_ant, dept_novo)
    VALUES (OLD.cd_funcionario, OLD.nm_funcionario,
OLD.nm_departamento, NEW.nm_departamento);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_troca_departamento
AFTER UPDATE OF nm_departamento ON DEPT_EMP
FOR EACH ROW
WHEN (OLD.nm_departamento IS DISTINCT FROM NEW.nm_departamento)
EXECUTE FUNCTION trg_troca_departamento_func();

```

Inserindo Dados para Teste

```

-- Para cadastro_cliente
INSERT INTO cadastro_cliente (nome_cliente, sobrenome, dt_cadastro)
VALUES
('João', 'Silva', CURRENT_DATE),
('Maria', 'Souza', CURRENT_DATE),
('Carlos', 'Pereira', CURRENT_DATE),
('Ana', 'Lima', CURRENT_DATE),
('Paula', 'Fernandes', CURRENT_DATE);

-- Para salario
INSERT INTO salario (nm_func, salario) VALUES
('MARCELO DINIZ', 1000),
('MARIO QUINELLO', 1000),
('CARLOS ROBERTO', 3000),
('JOSE SILVA', 5000),
('MARINA BEZERRA', 10000);

-- Para DEPT_EMP

```

```
INSERT INTO DEPT_EMP (CD_FUNCIONARIO, CD_FILIAL, NM_FUNCIONÁRIO,  
NM_DEPARTAMENTO) VALUES  
(1, 500, 'MARCELO DINIZ', 'TI'),  
(2, 500, 'CARLOS ROBERTO', 'FINANÇAS'),  
(3, 500, 'ROBERTO ALMEIDA', 'RH'),  
(4, 500, 'MARISA INACIO', 'VENDAS');
```