

```

-- CRIANDO A TABLE DE AUDITORIA
CREATE TABLE musica_auditoria (
    id SERIAL,
    idMusica    INTEGER NOT NULL,
    nomeMusica  VARCHAR(60) DEFAULT NULL,
    duracao     TIME DEFAULT NULL,
    dataevento  TIMESTAMP DEFAULT NULL,
    acao        VARCHAR(50) DEFAULT NULL,
    PRIMARY KEY (id)
);
-- CRIANDO A FUNÇÃO DO TIPO GATILHO
CREATE OR REPLACE FUNCTION auditoria_musica_delete()
RETURNS trigger AS $$
BEGIN
    INSERT INTO musica_auditoria( idMusica, nomeMusica, duracao, dataevento, acao)
    VALUES ( OLD . IdMusica, OLD.nomeMusica, OLD.duracao, NOW(), 'delete');
    RETURN NEW;
END;$$
LANGUAGE plpgsql;

-- CRIANDO O GATILHO COM A FUNÇÃO QUE DEVE SER EXECUTADA
CREATE TRIGGER musica_delete
BEFORE DELETE
ON musica
FOR EACH ROW
EXECUTE PROCEDURE auditoria_musica_delete();

-- TESTE
SELECT * FROM musica where idmusica = 2;
SELECT * FROM musica_auditoria;
SELECT * FROM musica_auditoria;
DELETE FROM musica where idmusica=2;

```

```

-- CRIANDO A TABLE DE AUDITORIA
CREATE TABLE musica_auditoria (
    id SERIAL,
    idMusica    INTEGER NOT NULL,
    nomeMusica  VARCHAR(60) DEFAULT NULL,
    duracao     TIME DEFAULT NULL,
    dataevento  TIMESTAMP DEFAULT NULL,
    acao        VARCHAR(50) DEFAULT NULL,
    PRIMARY KEY (id)
);
-- CRIANDO A FUNÇÃO DO TIPO GATILHO
CREATE OR REPLACE FUNCTION auditoria_musica_insert()
RETURNS trigger AS $$

```

```
BEGIN
    INSERT INTO musica_auditoria( idMusica, nomeMusica, duracao, dataevento, acao)
    VALUES ( NEW . IdMusica, NEW.nomeMusica, NEW.duracao, NOW(), 'insert');
    RETURN NEW;
END;$$
LANGUAGE plpgsql;

-- CRIANDO O GATILHO COM A FUNÇÃO QUE DEVE SER EXECUTADA
CREATE TRIGGER musica_insert
BEFORE INSERT
ON musica
FOR EACH ROW
EXECUTE PROCEDURE auditoria_musica_insert();

-- TESTE
SELECT * FROM musica_auditoria;
INSERT INTO musica (idmusica,nomemusica,duracao) values (91,'Rico', '00:03:55')
SELECT * FROM musica_auditoria;
```