

Explicação do código SQL

Gatilho 1

Todo formulário tem uma data limite. Caso a data atual ultrapasse a limite, a inserção de novas respostas por parte dos usuários não deve ocorrer.

```
CREATE OR REPLACE FUNCTION verificar_data_formulario()
RETURNS TRIGGER
AS
$$
DECLARE
    id_formulario INT;
    data_limite DATE;

BEGIN
    SELECT
        f.id_documento,
        f.data_limite
    INTO
        id_formulario,
        data_limite
    FROM formulario f
    JOIN pergunta p ON p.id_formulario = f.id_documento
    WHERE p.id = NEW.id_pergunta;

    IF (data_limite < CURRENT_DATE) THEN
        RAISE EXCEPTION
            'Formulário % não aceita mais respostas. Data limite: %',
            id_formulario,
            data_limite;
    END IF;

    RETURN NEW;
END
$$
LANGUAGE plpgsql;
```

Primeiramente, criamos uma função que retorna um objeto do tipo TRIGGER. Essa função verifica se a data limite foi ultrapassada e, caso tenha sido, lança uma exceção. É necessário fazer um SELECT para obter o id do formulário e sua data, pois não temos acesso direto a esses dados na tabela resposta.

```
CREATE TRIGGER verificar_data_formulario
BEFORE INSERT OR UPDATE ON resposta
FOR EACH ROW
EXECUTE PROCEDURE verificar_data_formulario();
```

Em seguida, criamos um gatilho que executa a função anterior sempre que um formulário for inserido ou atualizado.

Gatilho 2

Sempre quando um documento novo for criado, seu respectivo tipo de documento (apresentação, formulário, etc) também deve ser criado com valores padrão (seria como, por exemplo, criar um novo documento do Google Docs e ele já vir com configurações padrão de fonte, tamanho, etc).

```
CREATE OR REPLACE FUNCTION inicializar_documento()  
RETURNS TRIGGER  
AS  
$$  
BEGIN  
    CASE NEW.tipo  
        WHEN 'PLANILHA' THEN  
            INSERT INTO planilha (id_documento, largura_linhas, largura_colunas)  
            VALUES (NEW.id, 10, 10);  
  
        WHEN 'DOCUMENTO_TEXTO' THEN  
            INSERT INTO documento_texto (id_documento, fonte, tamanho_fonte)  
            VALUES (NEW.id, 'ARIAL', 12);  
  
        WHEN 'FORMULARIO' THEN  
            INSERT INTO formulario (id_documento, data_limite)  
            VALUES (NEW.id, CURRENT_DATE + INTERVAL '1' MONTH);  
  
        WHEN 'APRESENTACAO' THEN  
            INSERT INTO apresentacao (id_documento, objetivo)  
            VALUES (NEW.id, 'Apresentação de slides');  
    END CASE;  
  
    RAISE NOTICE 'Documento % inicializado', NEW.id;  
  
    RETURN NEW;  
END  
$$  
LANGUAGE plpgsql;
```

Primeiramente criamos uma função que retorna um objeto do tipo TRIGGER. Essa função verifica qual o tipo do documento inserido (planilha, documento de texto, formulário ou apresentação) e, de acordo com o tipo, insere um novo registro na tabela correspondente.

Ao final, também é lançada uma mensagem de log informando que o documento foi inicializado.