2.6 - Detalhe como você garantiria que a procedure e o trigger não introduzam vulnerabilidades de segurança, como SQL Injection.

Para garantir que a **procedure** e o **trigger** não introduzam vulnerabilidades de segurança, como **SQL Injection**, as melhores práticas incluem:

- **Evitar a concatenação de strings em SQL**: No caso da **procedure** e **trigger**, os valores são passados como parâmetros para a consulta SQL, o que evita a necessidade de concatenar strings e, assim, minimiza o risco de SQL Injection.
- **Usar funções e stored procedures**: Ao encapsular a lógica de inserção e auditoria em uma **procedure** ou **trigger**, elimina-se o risco de injeção de código SQL diretamente nas operações da aplicação.
- **Parâmetros fortemente tipados**: Ao usar tipos de dados como `bigint` e `numeric(15,2)` para os parâmetros, evitamos que entradas maliciosas possam ser interpretadas de maneira errada ou perigosas pelo banco de dados.