

1. Introdução às Triggers

Definição: Triggers são objetos do banco de dados que disparam uma ação automática quando certos eventos ocorrem em uma tabela.

Propósito:

- Automatizar tarefas repetitivas.
- Garantir a integridade e consistência dos dados.
- Auditar mudanças no banco de dados.

Quando usar:

- Validar dados antes de inserções ou atualizações.
- Manter colunas de auditoria como **last_modified**.
- Sincronizar dados entre tabelas.

2. Sintaxe Básica

sql

```
CREATE TRIGGER nome_da_trigger {BEFORE | AFTER} {INSERT | UPDATE | DELETE} ON nome_da_tabela FOR EACH ROW BEGIN -- corpo da trigger END;
```

Tipos de triggers:

- **BEFORE:** Executada antes do evento (inserção, atualização ou exclusão).
- **AFTER:** Executada após o evento.

Eventos:

- **INSERT:** Disparada ao inserir um novo registro.
- **UPDATE:** Disparada ao atualizar um registro existente.
- **DELETE:** Disparada ao excluir um registro.

3. Criando Triggers

Exemplo de Trigger BEFORE INSERT:

sql

```
CREATE TRIGGER validar_dados_antes_inserir BEFORE INSERT ON produtos FOR EACH ROW BEGIN IF NEW.preco < 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Preço não pode ser negativo'; END IF; END;
```

Exemplo de Trigger AFTER UPDATE:

sql

```
CREATE TRIGGER atualizar_data_modificacao AFTER UPDATE ON produtos FOR EACH ROW BEGIN SET NEW.last_updated = NOW(); END;
```

Exemplo de Trigger BEFORE DELETE:

sql

Copiar código

```
CREATE TRIGGER log_deletar_produto BEFORE DELETE ON produtos FOR EACH ROW BEGIN INSERT INTO log_deletar (produto_id, data) VALUES (OLD.id, NOW()); END;
```

4. Modificando e Excluindo Triggers

Modificando uma trigger:

- O MySQL não permite alterar diretamente uma trigger. É necessário excluir e recriar a trigger.

Excluindo uma trigger:

sql

```
DROP TRIGGER IF EXISTS nome_da_trigger;
```

5. Exemplos Práticos

Atualização automática de last_updated:

sql

```
CREATE TRIGGER atualizar_last_updated BEFORE UPDATE ON usuarios FOR EACH ROW BEGIN SET NEW.last_updated = NOW(); END;
```

Verificação de dados antes de inserção:

sql

```
CREATE TRIGGER verificar_idade BEFORE INSERT ON usuarios FOR EACH ROW BEGIN IF NEW.idade < 18 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Idade deve ser maior ou igual a 18 anos'; END IF; END;
```

Registro de alterações em uma tabela de log:

sql

```
CREATE TRIGGER log_alteracoes AFTER UPDATE ON usuarios FOR EACH ROW BEGIN INSERT INTO log_usuarios (usuario_id, acao, data) VALUES (NEW.id, 'UPDATE', NOW()); END;
```

6. Boas Práticas

- **Documentação:** Sempre documente suas triggers para facilitar a manutenção.
- **Performance:** Use triggers com cautela, pois podem impactar a performance do banco de dados.
- **Simples e claras:** Mantenha as triggers simples para facilitar o entendimento e manutenção.
- **Testes:** Teste exhaustivamente suas triggers para evitar comportamentos inesperados.