

Módulo 10: Triggers e Eventos

Módulo 10: Triggers e Eventos

Os triggers e eventos no MySQL são mecanismos que automatizam a execução de comandos e ações no banco de dados. Eles são úteis para monitorar e realizar tarefas automáticas com base em mudanças nos dados ou horários programados.

1. Triggers

O que são Triggers

Um trigger é um procedimento armazenado que é automaticamente executado (disparado) em resposta a eventos específicos em uma tabela, como inserções, atualizações ou exclusões.

- **Vantagens:**

- Automatizam processos.
 - Mantêm a integridade dos dados.
 - Executam ações baseadas em alterações nos dados.
-

Criação e Exclusão de Triggers

Sintaxe para Criação

```
CREATE TRIGGER nome_trigger
{BEFORE | AFTER} {INSERT | UPDATE | DELETE}
ON nome_da_tabela
FOR EACH ROW
corpo_do_trigger;
```

- **BEFORE:** O trigger é executado antes da operação.
- **AFTER:** O trigger é executado após a operação.
- **INSERT, UPDATE, DELETE:** Define o evento que dispara o trigger.

Exemplo Prático

Criar um trigger que insere um registro de log toda vez que um pedido for inserido:

```
CREATE TRIGGER log_pedido_inserido
AFTER INSERT ON pedidos
FOR EACH ROW
BEGIN
    INSERT INTO log_acoes (acao, data_hora)
    VALUES ('Pedido inserido', NOW());
END;
```

- Tabela **log_acoes**:
 - **acao** (texto da ação realizada).
 - **data_hora** (momento em que a ação ocorreu).

Exclusão de Triggers

Para excluir um trigger, use o comando **DROP TRIGGER**.

Exemplo: Excluir o trigger **log_pedido_inserido**:

```
DROP TRIGGER log_pedido_inserido;
```

Tipos de Triggers

1. **BEFORE**:
 - Executado antes da operação.
 - **Exemplo:** Validar dados antes de inserir.

```
CREATE TRIGGER valida_estoque
BEFORE INSERT ON pedidos
FOR EACH ROW
BEGIN
    IF NEW.quantidade <= 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'A quantidade deve ser maior que zero.';
    END IF;
END;
```

2. AFTER:

- Executado após a operação.
- **Exemplo:** Atualizar o estoque após uma venda.

```
CREATE TRIGGER atualiza_estoque
AFTER INSERT ON vendas
FOR EACH ROW
BEGIN
    UPDATE produtos
    SET estoque = estoque - NEW.quantidade
    WHERE id_produto = NEW.id_produto;
END;
```

2. Eventos

O que são Eventos no MySQL

Um **evento** é uma tarefa agendada que é executada automaticamente em intervalos regulares ou em horários específicos.

- **Vantagens:**
 - Automatizam tarefas repetitivas, como limpeza de logs ou backups.
 - Substituem cron jobs externos para tarefas relacionadas ao banco de dados.

Criação de Eventos Programados

Sintaxe Geral

```
CREATE EVENT nome_evento
ON SCHEDULE {AT data_hora | EVERY intervalo}
DO
comando_sql;
```

- **ON SCHEDULE:**
 - **AT:** Especifica uma data e hora exata.
 - **EVERY:** Define um intervalo regular.
- **DO:** Contém o comando SQL que será executado.

Exemplo Prático

1. Limpar Registros Antigos de Log

Criar um evento que apaga registros de log com mais de 30 dias, executado diariamente:

```
CREATE EVENT limpa_logs
ON SCHEDULE EVERY 1 DAY
DO
DELETE FROM log_acoes WHERE data_hora < NOW() - INTERVAL 30 DAY;
```

2. Enviar Relatórios Automáticos

Criar um evento para gerar um relatório de vendas todo primeiro dia do mês:

```
CREATE EVENT gera_relatorio_vendas
ON SCHEDULE EVERY 1 MONTH
STARTS '2024-01-01 00:00:00'
DO
INSERT INTO relatorios (descricao, data_geracao)
VALUES ('Relatório mensal de vendas', NOW());
```

Ativação e Desativação de Eventos

Por padrão, o agendador de eventos pode estar desativado. Ative-o com o comando:

```
SET GLOBAL event_scheduler = ON;
```

Verifique o status do agendador:

```
SHOW VARIABLES LIKE 'event_scheduler';
```

Aplicabilidade

1. Triggers:

- Automatizar atualizações em tabelas relacionadas.
- Monitorar alterações e gerar logs.
- Implementar validações automáticas.

Exemplo: Gerar logs automáticos ao excluir usuários.

```
CREATE TRIGGER log_usuario_excluido
AFTER DELETE ON usuarios
FOR EACH ROW
BEGIN
    INSERT INTO log_acoes (acao, data_hora)
    VALUES (CONCAT('Usuário excluído: ', OLD.nome), NOW());
END;
```

2. Eventos:

- Manutenção automatizada do banco de dados.
- Limpeza periódica de dados desnecessários.
- Geração de relatórios automáticos.

Exemplo: Um sistema de vendas que limpa carrinhos de compras abandonados após 24 horas.

```
CREATE EVENT limpa_carrinhos
ON SCHEDULE EVERY 1 DAY
DO
DELETE FROM carrinhos WHERE data_criacao < NOW() - INTERVAL 1 DAY;
```

Melhores Práticas

1. **Evitar Triggers Muito Complexos:**
 - Mantenha triggers simples para evitar impactos negativos na performance.
2. **Monitorar a Execução de Eventos:**
 - Teste eventos antes de ativá-los em ambientes de produção.
3. **Documentação:**
 - Registre claramente a finalidade de cada trigger ou evento.
4. **Planejamento:**
 - Use eventos para tarefas regulares que não exigem intervenção manual.

Com o uso correto de triggers e eventos, é possível automatizar grande parte das operações do banco de dados, melhorando a eficiência e a manutenção dos dados.

Vamos Praticar

Questões Práticas

1. Criação de um Trigger BEFORE

Crie um trigger que valide, antes de inserir um pedido na tabela `pedidos`, se a quantidade solicitada de um produto é menor ou igual ao estoque disponível.

2. Criação de um Trigger AFTER

Desenvolva um trigger que atualize o estoque na tabela `produtos` sempre que um novo pedido for registrado na tabela `pedido_produto`.

3. Registro de Logs

Crie um trigger que registre em uma tabela `log_acoes` toda vez que um registro for excluído da tabela `clientes`. O log deve incluir a data/hora e a identificação do cliente excluído.

4. Exclusão de um Trigger

Exclua o trigger que registra exclusões de clientes na tabela `log_acoes`.

5. Limpeza Automática com Eventos

Crie um evento que limpe registros da tabela `log_acoes` que tenham mais de 60 dias, executado diariamente às 23h59.

6. Geração Automática de Relatórios

Configure um evento que gere relatórios mensais de vendas, inserindo informações na tabela `relatorios` todo dia 1º de cada mês, à meia-noite.

7. Ativação do Agendador de Eventos

Ative o agendador de eventos no MySQL e verifique se ele está funcionando corretamente.

8. Modificação de um Trigger Existente

Altere um trigger existente para incluir a validação de um novo campo antes de realizar a inserção.

9. Teste de Performance de um Trigger

Execute um teste de performance em um trigger que é acionado frequentemente. Proponha melhorias para otimizar sua execução.

10. Documentação de um Evento

Documente detalhadamente um evento que gere relatórios automáticos, explicando sua finalidade, comportamento esperado e benefícios para o sistema.

Questões Teóricas

1. Conceito de Triggers

Explique o que são triggers no MySQL e cite pelo menos duas vantagens do seu uso.

2. Tipos de Triggers

Qual a diferença entre triggers `BEFORE` e `AFTER`? Cite exemplos de situações em que cada um seria utilizado.

3. Vantagens dos Eventos

Liste três vantagens de usar eventos no MySQL em vez de cron jobs externos.

4. **Validação com Triggers**
Como um trigger pode ser usado para validar dados antes de uma operação ser realizada? Explique com um exemplo teórico.
5. **Impacto de Triggers na Performance**
Quais são os cuidados a serem tomados ao criar triggers para evitar impactos negativos na performance do banco de dados?
6. **Automatização de Tarefas com Eventos**
Explique como eventos podem ser usados para automatizar a manutenção do banco de dados. Dê um exemplo prático.
7. **Ativação do Agendador de Eventos**
Por que o agendador de eventos pode estar desativado por padrão no MySQL?
Como ativá-lo?
8. **Exclusão de Triggers**
Qual é a sintaxe para excluir um trigger no MySQL? O que acontece com os dados afetados por triggers excluídos?
9. **Melhores Práticas para Triggers**
Quais são as melhores práticas para a criação de triggers? Explique como elas podem beneficiar a manutenção do banco de dados.
10. **Planejamento de Eventos**
Por que é importante planejar eventos no banco de dados? Cite situações em que um evento mal configurado pode gerar problemas.

Instruções para a Entrega das Atividades

1. **Elaboração e Envio do Arquivo**
 - Responda todas as questões de forma clara e objetiva.
 - Gere um arquivo no formato **.PDF** contendo as respostas de cada questão.
 - Envie o arquivo para os e-mails dos professores responsáveis.
2. **Validação da Atividade**
 - Após o envio do arquivo, procure o(s) professor(es) para realizar a validação da atividade.
 - **Não inicie a próxima atividade sem antes validar a anterior com o professor.**
3. **Forma de Validação**
 - **Explicação Verbal:** Explique cada resposta verbalmente ao(s) professor(es).
 - **Perguntas e Respostas:** Esteja preparado para responder aos questionamentos do(s) professor(es) sobre o conteúdo das respostas.
 - **Orientação:** Receba orientações sobre a apresentação do(s) tema(s).