

1. Conceito de Triggers

Triggers são procedimentos automáticos que são executados em resposta a eventos específicos no banco de dados, como inserções, atualizações ou exclusões.

Vantagens:

- Automação de regras de negócio;
 - Garantia de integridade e consistência dos dados.
-

2. Tipos de Triggers: BEFORE e AFTER

- **BEFORE:** executa antes da operação (INSERT, UPDATE, DELETE). Usado para validar ou modificar dados antes que sejam gravados.
- **AFTER:** executa após a operação, útil para ações como logging ou atualização de tabelas relacionadas.

Exemplo:

BEFORE INSERT para validar campos;

AFTER UPDATE para registrar alterações em histórico.

3. Vantagens dos Eventos sobre cron jobs externos

- Integração direta com o banco;
 - Execução garantida mesmo se servidor externo estiver indisponível;
 - Menor complexidade e dependência de ferramentas externas.
-

4. Validação com Triggers

Um trigger BEFORE INSERT pode validar se um valor está dentro de um intervalo esperado e abortar a inserção caso contrário, evitando dados inválidos.

5. Impacto de Triggers na Performance

Triggers complexos ou mal planejados podem retardar operações. Deve-se manter triggers simples, evitar loops e monitorar seu impacto.

6. Automatização com Eventos

Eventos podem agendar tarefas, como limpeza periódica de dados antigos.

Exemplo: evento que exclui registros com mais de 1 ano, rodando diariamente.

7. Ativação do Agendador de Eventos

O agendador pode estar desativado por padrão para evitar execução acidental de tarefas. Para ativar:

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

8. Exclusão de Triggers

Sintaxe:

```
DROP TRIGGER nome_do_trigger;
```

Após exclusão, o trigger deixa de ser executado, mas os dados existentes permanecem intactos.

9. Melhores Práticas para Triggers

- Manter triggers simples e rápidos;
 - Documentar a lógica;
 - Testar exaustivamente;
 - Evitar dependências complexas para facilitar manutenção.
-

10. Planejamento de Eventos

Eventos mal configurados podem causar execuções excessivas ou falhas, impactando performance e gerando dados incorretos. Planejamento evita conflitos e garante eficiência.