



Banco de Dados II

Prof. Vinícius Alves Hax



Na aula anterior

- Conceito de triggers
- Relacionando triggers com INSERT



Na aula de hoje

- Triggers com UPDATE e DELETE



Relembrando: Quando os triggers são acionados?

- Os triggers podem ser acionados ANTES da execução de um SQL ou APÓS
- Quando acionados antes usamos a palavra-chave BEFORE e quando depois usamos a palavra-chave AFTER

Relembrando: Criando um trigger

- Para criar um trigger são necessários dois passos
 - Criar a função que será associada ao trigger
 - Associar a função a um determinado evento (INSERT, DELETE, etc)

Exemplo de trigger com UPDATE

- Ver arquivo trigger3.sql
<https://tinyurl.com/ifsultrigger3>
- Crie os triggers e as funções
- Atualize um registro na tabela com o comando SQL abaixo

```
UPDATE teacher SET name = 'Mudei de nome!' WHERE id = 1;
```



O que acontece se na função chamada antes da atualização retornarmos OLD?



O que acontece se na função chamada antes do UPDATE retornarmos OLD?

R: A atualização não é feita pois é esperada que a função que atua antes do update retorne o registro pronto para salvar no banco de dados



DELETE trigger

- Assim como na função de um insert só temos a variável NEW, na função delete só temos a acesso à variável OLD
- A variável OLD é somente leitura. Não podemos modificar o registro

Exemplo de trigger com DELETE

- Ver arquivo trigger4.sql
<https://tinyurl.com/ifsultrigger4>
- Crie os triggers e as funções
- Delete um registro na tabela com o comando abaixo

```
DELETE FROM teacher WHERE id = 2;
```

Exceções

- Exceções são comportamentos diferentes do fluxo 'normal' de um código. Podem ocorrer por motivos diversos
 - Divisão por zero
 - Acessar arquivo não existente
- Algumas exceções já existem nas linguagens e bibliotecas, mas podemos criar novas

Ex: raise exception 'Registro não encontrado, não é possível deletar';



Exercícios

(Utilize para ambos as tabelas criadas na lista de exercícios 2)

Exercícios

- Crie um trigger que é chamado sempre que um registro da tabela teacher é atualizado. Esse trigger deve impedir que o nível novo seja menor do que o antigo.
- Crie um trigger que impeça que um professor seja deletado se existirem alunos vinculados a ele
- Crie um trigger que copie cada professor deletado para uma nova tabela (crie a tabela antes) chamada “deleted_teachers”



Referências

- <https://neon.tech/postgresql/postgresql-triggers>