



Prof. Taciano Balardin

www.taciano.pro.br

taciano@ulbra.edu.br

2015-2



Banco de Dados II

E-MAIL DE CONTATO:
taciano@ulbra.edu.br

SITE DA DISCIPLINA:
<http://www.taciano.pro.br/>



Triggers: Conceitos e Aplicabilidade

BANCO DE DADOS II

AULA 05

Triggers



- É muito comum, em aplicações que utilizam bancos de dados, que **ações sejam disparadas em resposta ou como consequência de outras**, realizando operações de cálculo, validações e, em geral, surtindo alterações na base de dados.
- Exemplo: Um sistema de vendas onde, a partir do registro de uma venda, é dado baixa no estoque de cada item.

Triggers



- Em muitos casos, os programadores optam por **executar** tais ações **a partir da própria aplicação**, executando várias instruções SQL em sequência para ter o resultado. Entretanto, **esta solução tende a tornar mais lenta** a execução de certas tarefas.
- Exemplo: Um sistema de vendas onde, a partir do registro de uma venda, é dada baixa no estoque de cada item.

Triggers



Uma TRIGGER é um objeto de banco de dados, associado a uma tabela, que permite a realização de processamentos em consequência de eventos.

- Quem são estes eventos?
 - São os comandos da DML: **INSERT**, **DELETE** ou **UPDATE**.
- Quantas TRIGGERS posso ter?
 - Podemos definir **inúmeras TRIGGERS** em uma base de dados, baseada em qual dos comandos acima irá dispará-la, sendo que, **para cada evento, podemos definir apenas uma TRIGGER**.
 - As TRIGGERS poderão ser disparadas para trabalharem **antes** ou **depois** do evento.





Triggers




- Os principais **pontos positivos** sobre as triggers são:
 - Parte do processamento que seria executado na aplicação passa para o banco, **poupando recursos da máquina cliente**.
 - **Facilita a manutenção**, sem que seja necessário alterar o código fonte da aplicação.
- Já os **pontos negativos** são:
 - Alguém que tenha **acesso não autorizado** ao banco de dados **poderá visualizar e alterar o processamento** realizado pelos gatilhos.
 - **Requer maior conhecimento** de manipulação do banco de dados (SQL) para realizar as operações internamente.

Exemplo

Nome:

Colunas:  Adicionar  Remover  Mover para cima  Mover para baixo

#	Nome	Tipo de dados	Tamanho/Itens	Unsigned	Permitir NULL	Zerofill	Padrão
 1	id	INT	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	nome	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão
3	estoque	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão

INSERT INTO produtos VALUES (1, 'Feijão', 10);

INSERT INTO produtos VALUES (2, 'Arroz', 5);

INSERT INTO produtos VALUES (3, 'Farinha', 15);

Exemplo

Nome:

itensvenda

Colunas:



Adicionar



Remover



Mover para cima



Mover para baixo

	#	Nome	Tipo de dados	Tamanho/Itens	Unsigned	Permitir NULL	Zerofill	Padrão
	1	venda_id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
	2	produto_id	INT	11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão
	3	qtd	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão

INSERT INTO itensvenda VALUES (1, 1, 3);

INSERT INTO itensvenda VALUES (1, 2, 1);

INSERT INTO itensvenda VALUES (1, 3, 5);

Sintaxe

CREATE

[DEFINER = { user | CURRENT_USER }]

TRIGGER trigger_name

trigger_time trigger_event →

trigger_name: nome da trigger

trigger_time: { BEFORE | AFTER } antes | depois

trigger_event: { INSERT | UPDATE | DELETE }

ON tbl_name **FOR EACH ROW**

tbl_name: tabela que a trigger ficará pendurada

BEGIN

trigger_body

END

OLD e NEW

- **OLD**: operador para ser utilizado no interior da TRIGGER para recuperar o valor anterior à um **UPDATE** ou **DELETE**;
- **NEW**: operador para ser utilizado no interior da TRIGGER para recuperar dados atuais a partir de comandos **INSERT** e **UPDATE**.

Exemplo

- Como atualizar automaticamente o estoque ao **inserir uma venda?**

BEGIN

```
UPDATE produtos SET estoque = estoque - NEW.qtd WHERE id = NEW.produto_id;
```

END

- Como atualizar automaticamente o estoque ao **excluir uma venda?**

BEGIN

```
UPDATE produtos SET estoque = estoque + OLD.qtd WHERE id = OLD.produto_id;
```

END

TRIGGERS

EXERCÍCIO

Utilizando as tabelas de *produtos* e *itensvenda*:

Criar uma trigger que vai **atualizar o valor do estoque** no momento em que um registro de venda for **ALTERADO** no banco de dados.

DESAFIO

