



DELETE, UPDATE, COMMIT, ROLLBACK

Prof. Dr. Nazareno de Oliveira Pacheco
nazareno.pacheco@prof.sc.senac.br

Apagando registros

- . É possível apagar registros de uma tabela utilizando o comando DELETE
- . A sintaxe deste comando é a seguinte:

```
DELETE FROM nome_tabela  
WHERE  
    condicoes;
```

Experimente

```
DELETE FROM produto  
WHERE  
    codigo = 512;
```

```
DELETE FROM produto  
WHERE  
    nome LIKE 'Tablet%'  
    AND fabricante = 'Coby';
```

Apagando registros

- O DELETE apagará todos os registros da tabela informada que atendam a determinadas condições
 - Estas condições devem ser informadas na cláusula WHERE, e funcionam da mesma maneira do SELECT
- **ATENÇÃO:**
 - Se a cláusula WHERE não for informada, TODOS os registros da tabela serão apagados!

Alterando registros

- Outra operação disponível no SQL é a alteração dos dados de registros já disponíveis em uma tabela
- Esta operação é chamada de UPDATE
- A sintaxe deste comando é a seguinte:

```
UPDATE nome_tabela  
SET  
    nome_col = novo_valor,  
    nome_outra_col = outro_novo_valor;  
WHERE  
    condicoes;
```

Experimente

```
UPDATE produto  
SET  
    preco = 260.00  
WHERE  
    codigo = 276;
```

```
UPDATE produto  
SET  
    fabricante = 'Hewlett-Packard',  
    preco = preco * 1.1  
WHERE  
    codigo IN (193, 194, 195, 196);
```

Recomendações sobre o UPDATE e DELETE

- . Tome muito cuidado com a cláusula WHERE
 - Uma falha pode apagar ou sobrescrever registros importantes!
 - Procure testar a cláusula WHERE realizando um SELECT * antes
 - Sempre que possível, utilize a chave primária da tabela como filtro, para garantir que apenas um registro seja alterado
 - Se possível, realize estas operações dentro de *transações* (veremos isso mais tarde)

Exercícios (15)

- 1 – Reajuste o preço de todas as câmeras em 7%
- 2 – Apague todos os produtos sem fabricante informado
- 3 – O nome do fabricante Hewlet~~t~~-Packard está escrito errado. Altere para Hewlett~~t~~-Packard em todos os registros
- 4 – Apague todos os Tablets da fabricante Coby
- 5 – Reajuse em 5% todos os televisores que estejam com preço abaixo da média. DICA: utilize dois SQLs: um para saber o preço médio, outro para reajustar os preços destes produtos)

Transações

- Suponha a seguinte tabela:

```
CREATE TABLE item_compra(  
    codigo    SERIAL NOT NULL PRIMARY KEY,  
    nome      VARCHAR(100) NOT NULL,  
    preco     NUMERIC(4,2)  
);
```

- Em muitas ocasiões, precisamos executar uma sequência de comandos para registrar uma operação:
 - Por exemplo, pode ser uma sequência de INSERTs que registram os produtos da nossa compra

Transações

- Em muitas ocasiões, precisamos executar uma sequência de comandos para registrar uma operação:
 - Por exemplo, pode ser uma sequência de INSERTs que registram os produtos da nossa compra

```
INSERT INTO item_compra (nome, preco)
VALUES ('Camisa', 40.0);
```

```
INSERT INTO item_compra (nome, preco)
VALUES ('Grava', 20.0);
```

```
INSERT INTO item_compra (nome, preco)
VALUES ('Calça', 105.0);
```

Transações

- Se um dos comandos falhar, nossa base de dados pode ficar comprometida
 - Neste exemplo, os itens cadastrados na base não vão corresponder ao que o usuário comprou

```
INSERT INTO item_compra (nome, preco)
VALUES ('Camisa', 40.0);
```

```
INSERT INTO item_compra (nome, preco)
VALUES ('Grava', 20.0);
```

```
INSERT INTO item_compra (nome, preco)
VALUES ('Calça', 105.0);
```

Transações

- . Em um caso como este, a melhor estratégia é a seguinte:
 - . OU a operação inteira é concluída e registrada na base
 - . OU a operação inteira é cancelada, e nada é registrado na base
- . Nestes casos, o ideal é executar nossa sequência de comandos dentro de uma *transação*

Transações

- Para iniciar uma transação, utilize o comando
`START TRANSACTION;`
- Enquanto a transação estiver ativa, todas as alterações (UPDATEs e DELETEs) serão registradas somente em memória, e vão ficar disponíveis somente na sessão do usuário atual:
 - Ou seja, nenhuma alteração será escrita na base de dados

COMMIT e ROLLBACK

- . Existem duas formas de finalizar uma transação:
- . COMMIT;
 - Este comando finaliza a transação e grava tudo o que foi realizado.
 - Deve ser utilizado quando todos os comandos da nossa transação foram executados com sucesso;

COMMIT e ROLLBACK

. ROLLBACK;

- Este comando joga fora todas as alterações realizadas dentro da nossa transação, ou seja, nenhuma das alterações fica gravada.
- Deve ser utilizado quando alguma coisa falhou em nossa transação e queremos cancelá-la

Senac

VIDA DE PROGRAMADOR

.COM.BR

/* HISTÓRIA REAL
ENVIADA POR
GUSTAVO ACHKAR */



#270

POOTZ, DELETEI TODAS
AS LINHAS DE UMA TABELA
DO BANCO DE DADOS!



A TABELA
"CLIENTE"

CARAMBA! MAS
VOCÊ JÁ DEU
COMMIT?



AH TÁ, PERAÍ...

commit

PLOFT!

