Atualizado em 11/03/2019

Atualização de Dados via SQL-DML

Alex Sandro
alex@ifpb.edu.br

Sumário

- *Comando INSERT*
  - *Inclusão simples, inclusão de dados com SELECT, inclusão de dados com valores **default***
- *Comando UPDATE*
  - *Alteração simples, alteração usando subquery*
- *Comando DELETE*
  - *Eliminação simples, eliminação utilizando subquery*
- *Comando TRUNCATE TABLE*
  - *Remoção de todos os registros de uma tabela*

Comando INSERT

- ***Propósito***
  - *O comando INSERT **inclui** um registro no final de uma tabela de acordo com os valores especificados*
- ***Sintaxe***

  INSERT INTO *<tabela> [(<lista de campos>)]*
  VALUES *<lista de valores>*
- ***Argumentos***
  - ***<tabela>:*** *especifica a tabela na qual o registro será incluído*
  - ***<lista de campos>:*** *especifica os campos da tabela cujos valores serão inseridos*

Inclusão simples

- ***Argumentos (cont.)***
  - ***VALUES <lista de valores>:*** *especifica os valores (correspondentes aos respectivos campos) a serem inseridos na tabela*
- ***Exemplo1:*** *inserir na tabela* ***si_usuario*** *os valores: login=claudia, senha=claudia, matricula=20191001 e nome=Claudia Cruz*

```
INSERT INTO si_usuario (login,
senha,matricula,nome) VALUES
('claudia','claudia','20191001','Clau
dia Cruz')
```

Inclusão simples

- ***Exemplo 2:*** *Inserir na tabela de* ***setores*** *o setor de Contabilidade, cuja sigla é CON, o ramal é 108, o setor superior é Finanças (FIN) e o código do chefe do setor é 9*

```
INSERT INTO setor (sigla, ramal, nome,
ramal, superior, chefe)
VALUES ('CON', 108, 'Contabilidade',
9, 101) -- 101 é o código do chefe
```

- ***Exemplo 3:*** *Inserir um registro na tabela* ***si_destinatario****, cujo idMens=5 e recebedor=alex*

```
INSERT si_recebedor VALUES (5,'alex')
```

Incluindo dados com SELECT

- ***Propósito***
  - *Inserir dados em uma tabela tendo como valores o resultado de um SELECT*
- ***Sintaxe***

  INSERT INTO *<tabela>*
  SELECT <lista de campos> FROM <tabela>
  [WHERE, GROUP BY, ORDER BY, etc.]
- ***Observações***
  - *As colunas resultantes do SELECT têm que combinar com os tipos dos dados das colunas da tabela na cláudula INTO*
  - *Não se usa o **VALUES** no **INSERT** (neste caso)*

Incluindo dados com SELECT

```
IF EXISTS(SELECT 1 FROM
  information_schema.tables WHERE table_name
  = 'tabela1')
   DROP TABLE tabela1


CREATE TABLE tabela1(
  coluna1 integer,
  coluna2 varchar(40) NULL,
  coluna3 char(1) NULL,
  coluna4 varchar(30) NULL
);
INSERT INTO tabela1
SELECT nome, sexo, endereco
FROM funcionário
WHERE estcivil <> 'C'
```

Incluindo dados com SELECT

```
DROP TABLE tabela1;


CREATE TABLE tabela1 (
  pedido smallint CONSTRAINT PK_tabela
  PRIMARY KEY,
  cliente varchar(40) NULL,
  vendedor varchar(40) NULL
)
INSERT INTO tabela1
  SELECT p.código, c.nome, f.nome
  FROM pedido p JOIN cliente c
  ON c.código = p.cliente
  JOIN funcionário f
  ON f.código = p.vendedor
```

Incluindo dados com valor default

```
DROP TABLE tabela1


CREATE TABLE tabela1(
   coluna1 int SERIAL,
   coluna2 varchar(40) NOT NULL
   CONSTRAINT DF_col2_tab1 DEFAULT ('valor
   default'),
   coluna3 int NULL,
   coluna4 varchar(30) NOT NULL
)
1. INSERT INTO tabela1 (coluna4) VALUES
   ('valor 1-4')
2. INSERT INTO tabela1 (coluna2,coluna4)
   VALUES ('valor 2-2','valor 2-4')
3. INSERT INTO tabela1 (coluna2,coluna3,coluna4)
   VALUES ('valor 3-2',33,'valor 3-4')
```

Comando UPDATE

- **Propósito**
  - *O comando UPDATE **altera** os valores armazenados no(s) registro(s) de uma tabela*
- **Sintaxe**

```
UPDATE  <tabela>
SET <coluna1=valor1 [,coluna2=valor2,…,
coluna_n=valor_n]>
[WHERE <condição >]
```

- **Argumentos**
  - ***<tabela>:** especifica a tabela na qual o(s) registro(s) será(ão) alterados*
  - ***SET <coluna1=valor1[,coluna2=valor2,…]>:** especifica as colunas a serem alteradas e seus novos valores*

Alteração simples

- ***Argumentos (cont.)***
  - ***WHERE <condição>:*** *especifica as condições que precisam ser satisfeitas pelos registros que terão seus valores atualizados*
- ***Exemplo 4:*** *Aumente o salário dos funcionários casados em 25%*

```
UPDATE funcionario
SET salario = salario*1.25
WHERE estadoCivil = 'C'
```

Alteração simples

- ***Exemplo 5:*** *Conceder um aumento de 15% no salário dos funcionários homens, casados ou divorciados e que tenham nascido entre 01/01/50 e 31/12/70.*

```
UPDATE funcionario
SET salario=salario*1.15
WHERE sexo='M' AND
      estCivil IN ('C','D') AND
      dataNasc BETWEEN '1950/01/01'
               AND '1970/12/21'
```

Alteração simples

- ***Observação importante:*** *Após inserir um determinado registro em uma tabela, se algum campo não receber valor, terá como informação NULL. Após a atualização do determinado campo, o mesmo não será mais NULL.*

```
UPDATE funcionario
SET salario=salario*0.95


UPDATE funcionario
SET salario = NULL
WHERE codigo = 1
```

Alteração utilizando subquery

- ***Exemplo 6:*** *Fornecer um aumento de 18% no salário dos funcionários que tenham o salário menor que a média de salários da empresa.*

```
UPDATE funcionario
SET salario=salario*1.18 WHERE salario
< (SELECT AVG(salario) FROM
funcionario)
```

- ***Exemplo 7:****Baixar o preço em 12% dos produtos com preço de venda igual ao do produto mais caro vendido na empresa.*

```
UPDATE produto SET venda=venda*0.88
WHERE venda = (SELECT MAX(venda) FROM
produto)
```

Alteração simples

- ***Exemplo 8:*** *Alterar de todos os funcionários a função 'Analista Junior' para 'Analista Senior'.*

```
UPDATE funcionario
SET funcao = (SELECT codigo FROM
funcao WHERE nome = 'Analista
Senior')
WHERE funcao = (SELECT codigo FROM
funcao WHERE nome = 'Analista
Junior')
```

Comando DELETE

- ***Propósito***
  - *O comando DELETE **elimina** registros de uma tabela de acordo com a condição especificada.*
- ***Sintaxe***

```
DELETE FROM  <tabela>
[WHERE <condição >]
```

- ***Argumentos***
  - ***<tabela>:** especifica a tabela na qual o(s) registro(s) será(ão) excluídos*
  - ***WHERE <condição>:** especifica as condições que precisam ser satisfeitas para a remoção dos registros*

Remoção simples

- ***Exemplo 9:*** *Remova todos os registros da tabela 'usuario'.*

```
DELETE FROM usuario
```

- ***Exemplo 10****: Excluir todos os clientes do tipo pessoa física que não possuam telefone*

```
DELETE FROM cliente WHERE tipo='F'
AND fone IS NULL
```

- ***Exemplo 11****: Excluir todos os alunos desistentes do curso de Banco de Dados Avançado*

```
DELETE FROM aluno WHERE situacao='D'
AND curso='BDA'
```

Eliminação usando subquery

- ***Exemplo 12:*** *Excluir todos os atletas cujo esporte praticado 'basquete'*

```
DELETE FROM atleta
WHERE codEsporte = (SELECT codigo
FROM esporte WHERE nome = 'basquete')
```

- ***Exemplo 13****: Excluir todos os pedidos dos clientes dos estados da Paraíba e Pernambuco.*

```
DELETE FROM pedido WHERE cliente =
(SELECT codigo FROM cliente  WHERE
cidade IN (SELECT codCid FROM cidade
WHERE uf IN ('PB','PE')))
```

Comando TRUNCATE TABLE

- ***Propósito***
  - *Comando usado para **excluir** todas as linhas de uma tabela sem percorrê-la*
- ***Sintaxe***

  TRUNCATE TABLE *<tabela>*
- ***Características***
  - *Semelhante ao DELETE FROM sem o WHERE*
  - *Quase sempre é mais rápido que o DELETE, especialmente em tabelas grandes*
  - *TRUNCATE TABLE não salva informações no Log de Transações*

Bibliografia

- ***PostgreSQL Tutorial.*** *Disnponívem em: http://www.postgresqltutorial.com/*