

DATA MANIPULATION LANGUAGE - DML

Uma vez criada a estrutura física do banco de dados é necessário popular as tabelas. Para isso utilizam-se os comandos da DML (*Data Manipulation Language*). Nesta seara encontramos comandos como *Insert*, *Update*, *Delete*, *Commit* e *Rollback*.

DML(Linguagem de Manipulação de Dados):

Permite a inclusão, remoção, seleção ou atualização de dados armazenados no banco de dados; Controle de acesso, permitindo proteção dos dados de manipulações não autorizadas;

O Índice é um arquivo auxiliar associado a uma Tabela. Sua função é acelerar o tempo de acesso às linhas de uma Tabela, cria ponteiros para os dados armazenados em colunas específicas. O Banco de dados usa o Índice de maneira semelhante ao índice remissivo de um livro, verifica um determinado assunto no Índice e depois localiza a sua posição em uma determinada página.

3.1 INSERT

Comando responsável por adicionar um ou mais registros na tabela de Banco de Dados. Os campos que forem omitidos recebem valores NULOS (“NULL”).

Sintaxe:

```
INSERT [TRANSACTION transaction] INTO <object> [(col [, col ...])]
{VALUES (<val> [, <val> ...)] | <select_expr>};
<object> = tablename | viewname
<val> = { :variable | <constant> | <expr>
| <function> | udf ([<val> [, <val> ...]])
| NULL | USER | RDB$DB_KEY | ?
}
```

Exemplo:

```
INSERT INTO clientes (id, nome) VALUES (1, 'Nome do Cliente');
```

NOTE BEM:

- ✓ A lista de colunas é opcional no comando *Insert*;
- ✓ Caso não seja definida, assume-se a seqüência da criação da tabela;
- ✓ Para maior clareza do código e evitar erros em caso de reestruturação da tabela recomenda-se utilizar a lista de colunas.
 - ✓ Valores alfanuméricos devem ser especificados entre aspas simples 'exemplo';
 - ✓ Valores numéricos não devem conter esse separador.
 - ✓ Valores do tipo data devem ser tratados de acordo com as especificações do SGBD;
 - ✓ Ao incluir uma linha no Banco de Dados, o gerenciador checará as restrições de integridade (*Constraints*). Assim chaves primárias, estrangeiras, domínios, etc são checadas no momento da inclusão.
 - ✓ Caso alguma restrição for violada, a linha não será incluída e uma mensagem será emitida pelo SGBD.
 - ✓ As colunas que não tiverem valores atribuídos terão conteúdo *NULL* ou *default*, caso não haja a restrição *NOT NULL*.
 - ✓ Para inserir um valor nulo deve informar *NULL* no lugar correspondente a coluna.

3.1.1 Inserindo Várias Linhas

Para incluir diversas linhas em uma tabela, utiliza-se o comando *INSERT* em conjunto com o comando *SELECT*. Isso irá copiar as linhas de uma tabela para outra; A lista de colunas do *SELECT* deve corresponder a totalidade de colunas da tabela.

Exemplo:

```
INSERT INTO vendas_old SELECT * FROM vendas WHERE data_venda =
CORRENTE DATE;
```

Outra forma é usando o comando *INSERT* por várias vezes consecutivas acompanhado do finalizador ponto-e-virgula . Ou também após o comando *VALUES* colocando os dados entre parênteses separados por vírgula.

Exemplo:

```
INSERT INTO vendas_old (prod_id, prov_valor) VALUES (1, 2.00),
(2, 3.00), (3, 5.00), (4, 10.00);
```

3.1.2 Laboratório

```
INSERT INTO autor (aut_id, aut_nome) VALUES (1, 'Renato Russo');
INSERT INTO autor (aut_id, aut_nome) VALUES (2, 'Ton Jobim');
INSERT INTO musica (mus_id, mus_nome, mus_duracao) VALUES (1,
'sera', '00:02:28');
INSERT INTO musica (mus_id, mus_nome, mus_duracao) VALUES (2, 'Ainda
é Cedo', '00:03:55');
INSERT INTO gravadora (grav_id, grav_nome, grav_tel, grav_contato)
VALUES (1, 'EMI', '1122334455', 'José Vitor');
INSERT INTO cd(cd_id, cd_nome, cd_preco, dt_lancamento, grav_id)
VALUES (1, 'Mais do Mesmo', 20.50, '21.12.1998', 1);
```

3.1.3 Exercícios

a) Popule as tabelas do banco de dados conforme representado abaixo:

Gravadora

| Grav_id | Grav_nome | Grav_tel | Grav_contato |
|---------|-----------|------------|--------------|
| 1 | EMI | 1122334455 | José Vitor |
| 2 | Poligram | 3688790000 | Joaquim |
| 3 | Sony | 5645457777 | Pedro |
| 4 | Globo | 1132324444 | Josefina |

Mus_autor

| mus_id | aut_id |
|--------|--------|
| 1 | 1 |
| 2 | 1 |
| 3 | 3 |
| 4 | 3 |
| 5 | 2 |

CD

| cd_id | cd_nome | cd_preco_venda | cd_dt_lancamento | grav_id |
|-------|-------------------|----------------|------------------|---------|
| 1 | Mais do mesmo | 20.50 | 21.12.98 | 1 |
| 2 | Bate Boca | 19.00 | 12.12.79 | 1 |
| 3 | Perfil | 17.50 | 23.06.99 | 3 |
| 4 | Essa Mulher | 23.10 | 01.07.85 | 2 |
| 5 | Perfil 2 | 18.00 | 20.02.2001 | 3 |
| 6 | Grande Encontro 1 | 30.00 | 30.12.2002 | 2 |

Autor

| aut_id | Aut_nome |
|--------|--------------|
| 1 | Renato Russo |
| 2 | Ton Jobim |
| 3 | Elis Regina |

Música

| mus_id | mus_nome | mus_duracao |
|--------|---------------|-------------|
| 1 | Será | 00:02:00 |
| 2 | Ainda é Cedo | 00:03:02 |
| 3 | A Violeira | 00:02:30 |
| 4 | Anos Dourados | 00:02:11 |
| 5 | Imagina | 00:01:50 |

Item_cd

| cd_id | mus_id | Item_faixa |
|-------|--------|------------|
| 1 | 1 | 1 |
| 1 | 2 | 2 |
| 2 | 5 | 6 |

ATENÇÃO:

Observe a sequência de inserção dos dados.

3.2 UPDATE

Comando responsável alteração de ou um mais registros na tabela de Banco de Dados.

Sintaxe:

```
UPDATE [TRANSACTION transaction] {table | view}
SET col = <val> [, col = <val> ...]
[WHERE <search_condition> | WHERE CURRENT OF cursor];
```

Exemplo:

```
UPDATE CLIENTE SET DATA_INCLUSAO = CURRENT DATE;
```

3.2.1 Where

Comando responsável por especificar qual registro da tabela será alterado quando usado em conjunto com a cláusula UPDATE.

NOTE BEM:

- ✓ O conteúdo a ser atualizado deve respeitar o tipo de dado da coluna.
- ✓ A condição expressa no comando servirá para definir quais linhas devem ser atualizadas.
- ✓ Caso a cláusula WHERE contenha a chave primária somente uma linha da será atualizada.
- ✓ Todas as restrições de integridade (*Constraints*) serão avaliadas pelo SGBD.

3.2.2 Laboratório

Se quisermos alterar o preço de um CD, podemos utilizar o seguinte comando filtrando na cláusula WHERE a chave primária da tabela CD:

Exemplo:

```
UPDATE cd SET cd_preco = 21,00 WHERE cd_id = 1;
```

Se o aumento for para todos os cds de uma determinada gravadora, utilizaremos como filtro da cláusula WHERE grav_id:

Exemplo:

```
UPDATE cd SET cd_preco = 22,00 WHERE grav_id = 1;
```

Imagine que ocorra um aumento de preço generalizado para todos os CDs em 5%. Assim multiplica-se o próprio preço do CD por 1,05.

Exemplo:

```
UPDATE cd SET cd_preco = cd_preco * 1,05;
```

Outros Exemplos:

```
UPDATE cd SET cd_preco = 15 WHERE cd_id = 1;
UPDATE autor SET aut_nome = 'B. Manilow' WHERE aut_id = 2;
UPDATE cd SET cd_preco = cd_preco * 1,10 WHERE grav_id = 3;
UPDATE gravadora SET grav_nome = 'E.M.I', grav_tel = '2144332211'
WHERE grav_id = 1;
```

3.3 DELETE

Comando responsável pela exclusão de ou um mais registros na tabela de Banco de Dados.

Sintaxe:

```
DELETE [TRANSACTION transacional] FROM table
{[WHERE <search_condition>] | WHERE CURRENT OF cursor};
```

Exemplo:

```
DELETE FROM VENDAS WHERE DATA_VENDA <= '1-JAN-1999';
```

3.3.1 Where

Comando responsável por especificar qual registro da tabela será excluído quando usado em conjunto com a cláusula DELETE.

NOTE BEM:

- ✓ O comando DELETE pode, da mesma forma que o comando UPDATE, afetar uma ou mais linhas de uma tabela. Desta forma o escopo da exclusão será definido pela cláusula WHERE.
- ✓ Se for especificado na cláusula WHERE a chave primária, somente uma linha será excluída.
- ✓ Antes de uma exclusão, o gerenciador sempre fará uma checagem na integridade referencial.

3.4 CONTROLE BÁSICO DE TRANSAÇÕES

Uma vez realizadas as operações de inclusão, alteração e exclusão, a modificação realizada não estará ainda no banco de dados. Ela é visível apenas na seção atual.

Para que os comandos de *Insert*, *Update* e *Delete* sejam enviados para o banco é necessário utilizar os comandos **COMMIT** ou descartados pelo comando **ROLLBACK**.

3.5 LABORATÓRIO

Neste primeiro exemplo, apenas o autor com código 1 será excluído (desde que não haja relacionamentos com outras tabelas):

Exemplo:

```
DELETE FROM autor WHERE aut_id = 1;
```

Neste caso todos os CD's da gravadora 2 serão excluídos:

Exemplo:

```
DELETE FROM cd WHERE grav_id = 2;
```