

[◀ VOLTAR](#)

Manipulação de dados nas tabelas - inserindo dados - parte I

Utilizar comandos para manipulação de dados (insert), com o propósito de inserir novos registros em uma ou mais tabelas, a partir do uso de comandos DML.

NESTE TÓPICO

[Marcar tópico](#)

Inserindo novos registros em uma tabela

Os comandos do subconjunto DML (*Data Manipulation Language*), ou Linguagem de Manipulação de Dados, trabalham os registros (dados) de um banco de dados e têm por finalidade criar novas linhas, atualizar o que já foi criado e eliminar qualquer registro que já exista na tabela. Nesta aula aprenderemos a trabalhar com dados diretamente no SQL PLUS, ferramenta do SGBDR Oracle. Mas, antes, precisamos lembrar algumas informações. Abaixo, seguem os tipos de dados usados:

Tipos de dados		Descrição
Char(n)		Cadeia de caracteres de tamanho fixo – n – representa o tamanho, máximo de 255 dígitos.
Varchar(n) Varchar2(n)	ou	Cadeia de caracteres de tamanho variável – n – representa o tamanho, máximo de 4000 dígitos.
Long		Cadeia de caracteres de tamanho variável – n – representa o tamanho, máximo de 2Gb dígitos. Apenas um tipo deste por tabela.
Raw e longraw		Dados binários, armazenamento de sons e imagens.

Number(x) ou Valores numéricos inteiros e reais, onde x representa a parte inteira, Number(x,y) enquanto y representa as casas decimais. Tamanho de até 38 posições.

Date Armazena data e hora.

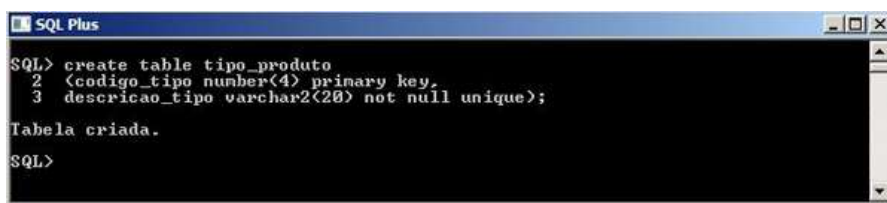
É necessário, também, conhecer a estrutura de uma tabela e saber interpretar sua leitura. Por exemplo, para visualizar a estrutura de uma tabela, utilizamos o seguinte comando: desc nome da tabela. Exemplo:

1. Desc cliente

Vamos analisar a resultado da instrução a seguir:

1. create table tipo_produto
2. (codigo_tipo number(4) primary key,
3. descricao_tipo varchar2(20) not null unique);

Resultado da criação:



```
SQL> create table tipo_produto
2 (codigo_tipo number(4) primary key,
3 descricao_tipo varchar2(20) not null unique);

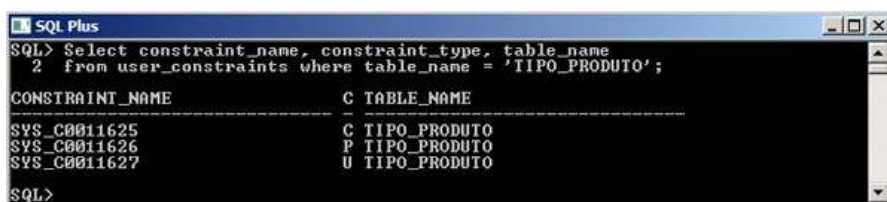
Tabela criada.

SQL>
```

Após análise do resultado acima, é possível notar a sequência de criação da estrutura, isto é muito importante, pois em determinadas instruções devemos obedecer a essa sequência de criação. Outra informação importante refere-se às regras existentes na estrutura que devemos seguir, para isso, se houver dúvidas em relação às regras ou também chamadas *constraints*, deve-se utilizar o seguinte comando de visualização das regras:

1. Select constraint_name, constraint_type, table_name
2. from user_constraints where table_name = 'TIPO_PRODUTO';

Resultado da visualização das regras de uma tabela:



```
SQL> Select constraint_name, constraint_type, table_name
2 from user_constraints where table_name = 'TIPO_PRODUTO';

CONSTRAINT_NAME          C TABLE_NAME
-----
SYS_C0011625              C TIPO_PRODUTO
SYS_C0011626              P TIPO_PRODUTO
SYS_C0011627              U TIPO_PRODUTO

SQL>
```

Depois deste comando, saberemos quais são as regras existentes nas colunas de uma tabela. Para visualizar em outra tabela, altere o final do comando trocando apenas o nome da tabela, digite em MAIÚSCULO.

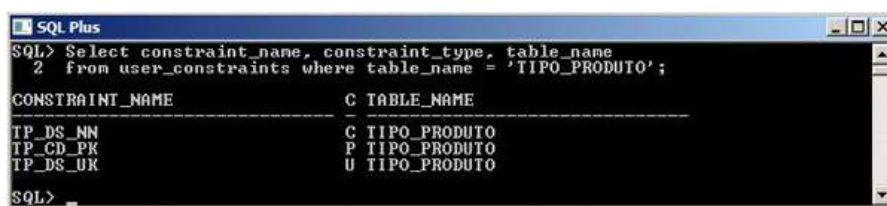
Os tipos de *constraints* são representados por letras:

- P ? Primary Key ? Chave primária
- R ? References ? Chave estrangeira
- C ? Not null ? Não nulo ou preenchimento obrigatório
- U ? Unique key ? Valores únicos
- C ? Check ? Validação (semelhante ao not null)

Para que não haja problemas com a criação das regras, a sugestão é sempre criar a estrutura de uma tabela usando o comando que permite dar nomes às regras, como exibido a seguir:

1. create table tipo_produto
2. (codigo_tipo number(4) constraint tp_cd_pk primary key,
3. descricao_tipo varchar2(20) constraint tp_ds_nn not null
4. constraint tp_ds_uk unique);

Observe o resultado do comando abaixo após o uso da instrução acima:



```
SQL> Select constraint_name, constraint_type, table_name
2  from user_constraints where table_name = 'TIPO_PRODUTO';
```

CONSTRAINT_NAME	C	TABLE_NAME
TP_DS_NN	C	TIPO_PRODUTO
TP_CD_PK	P	TIPO_PRODUTO
TP_DS_UK	U	TIPO_PRODUTO

```
SQL>
```

1. Select constraint_name, constraint_type, table_name
2. from user_constraints where table_name = 'TIPO_PRODUTO';

Agora ficou mais simples a visualização das regras:

TP_CD_PK - representa a criação de uma regra na tabela tipo_produto, onde TP é a abreviação do nome da tabela, CD o nome da coluna e PK a regra que está sendo criada, a primary key e, ao seu lado, a letra P que representa a primary key.

Após conhecer a tabela e suas regras, podemos inserir dados na tabela.

Sintaxe do comando que permite a inserção de dados em uma tabela:

1. Insert into nome da tabela values (valor coluna1,..., valor colunaN);

Nesta versão de comando é necessário seguir a ordem de criação da estrutura da tabela e suas regras.

Exemplo:

1. Insert into tipo_produto values (1,'Computador');



```
SQL> insert into tipo_produto values (1,'Computador');
1 linha criada.
SQL>
```

Note que os tipos de dados: char, varchar ou varchar2 e date, para inserção e posteriormente pesquisa, deverão estar entre apóstrofes. O campo do tipo numérico é opcional e, por ser tratar de software em língua inglesa, usa-se ponto como separador de casas decimais, não se utiliza vírgula e separadores de milhar. Exemplos: 5000 ? 500.35 ? 100.00

Para visualizar o resultado da execução na tabela, usa-se o comando de criação de relatórios, que, adiante, será apresentado com detalhes:

1. Select * from tipo_produto;



```
SQL> select * from tipo_produto;
CODIGO_TIPO DESCRICAO_TIPO
-----
1 Computador
SQL>
```

Este comando realiza uma pesquisa na tabela, o asterisco representa todas as colunas existentes.

Como a linguagem SQL é muito versátil, é possível utilizar outra versão deste comando DML para inserção de dados:

1. Insert into nome tabela (coluna1,...,colunaN) values (valor coluna1,..., valor colunaN);

Onde: após o nome da tabela é digitado o nome da(s) coluna(s) e não há necessidade de seguir a ordem de criação da tabela, mas os dados devem seguir a ordem das colunas existentes no comando, por exemplo:

Tabela Empregado:

Cod_emp	Numérico	4	Pk
Nome_emp	Alfanumérico	10	Nn

Dt_nasc	Data
Salario	Numérico 6,2

Inserindo dados sem seguir a ordem da estrutura da tabela:

1. Insert into empregado (nome_emp, salario, cod_emp, dt_nasc)
2. Values ('Mônica',7509.35,12,'10/12/1977');

Nulo é algo, um dado que não possui representação na tabela de símbolos da informática chamada ASCII, ou é somente representado por vazio. Para esta situação existe uma maneira para realizar instrução de inclusão de dados.

Para que um nulo seja inserido, a coluna não pode ser uma chave primária ou de preenchimento obrigatório, caso represente um dos casos não há como deixá-la vazia, devido aos princípios de cada regra.

Suponha que na "tabela empregado" (citada acima), no momento da inclusão de valores ou dados para um dos empregados, não tenha o valor do salário, neste caso, a instrução de inserção poderia ser realizada da seguinte forma:

1. Insert into empregado (nome_emp, salario, cod_emp, dt_nas)
2. Values ('Mônica',null,12,'10/12/1977');

Observe que a coluna "salário" está na instrução, mas no momento de inclusão está à palavra *null* sem apóstrofe, esta sintaxe permite a não inclusão de dados.

A instrução pode ser realizada da seguinte forma:

1. Insert into empregado (nome_emp, cod_emp, dt_nas)
2. Values ('Mônica',12,'10/12/1977');

Como nesta instrução não foi inserida a coluna "salário", não há necessidade de colocar a palavra *null*.

Outra situação:

1. Insert into empregado values (12, 'Mônica','10/12/1977',null);

Observe que utilizando a primeira versão sem as colunas e obedecendo à estrutura, será obrigatório o uso da palavra *null*.

Referências

BEIGHLEY, Lynn. *Use a Cabeça SQL*. Rio de Janeiro: Alta Books, 2008.

FANDERUFF, Damaris. *Dominando o Oracle 9i: Modelagem e Desenvolvimento*, São Paulo: Makron, 2003.

GRAVES, Mark. *Projeto de banco de dados com XML*. São Paulo: Pearson, 2003.

MORELLI, Eduardo Terra. *Oracle 9i Fundamental: SQL, PL/SQL e Administração*, São Paulo, Editora Érica, 2002.

PRICE, Jason. *Oracle Database 11g SQL*. (tradução: João Eduardo Nóbrega Tortello). Porto Alegre: Bookman, 2009.

SILVA, Robson. *Oracle Database 10g Express Edition*. São Paulo: Editora Érica, 2007.



Avalie este tópico



ANTERIOR

Alteração de estrutura de uma tabela

Biblioteca

(<https://www.uninove.br/conheca->

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Manipulação de dados nas tabelas - inserindo

dados - parte

© Todos os direitos reservados

Ajuda?

PRÓXIMO

([https://ava.un](https://ava.uninove.br/seu/AVA/topico/topico.php)

idos=)

