

[◀ VOLTAR](#)

Trabalhando com a estrutura de tabelas

Apresentação da criação das estruturas das tabelas e regras, a partir do uso de comandos DDL.

NESTE TÓPICO

[> Criando Tabelas](#)[> Criando Tabelas](#)[> Sintaxe:](#)[> Referências](#)[Marcar
tópico](#)

Criando Tabelas

Para a criação de tabelas são utilizados os comandos DDL do subconjunto de comandos SQL.

DDL – *Data Definition Language* – Linguagem de definição de dados.

Conjunto de comandos que trabalham a estrutura de uma tabela.

Regras para nomear tabelas, colunas e *constraint*:

- Não começar com número;
- Não possuir espaço em branco entre as palavras;
- Não usar acentuação, nem caracteres especiais;
- Diferenciar o uso de letras maiúsculas ou minúsculas;
- A digitação pode ser feita em uma única linha ou em várias linhas.

Esta parte de desenvolvimento é chamada de modelo físico, onde após ser criado o MER, há a transformação em modelo relacional/ lógico, a última fase é criação prática em um SGDBR, em nosso caso, o Oracle.

Criando Tabelas

Sintaxe:

```
create table nome_tabela
(nome_coluna1 tipo_dados (tamanho)
[constraint nome_constraint tipo_constraint],...,
nome_colunaN tipo_dados (tamanho)
[constraint nome_constraint tipo_constraint]);
```

Exemplo 1: Utilizando o código SQL para criar a tabela: (versão completa do comando).

Criando a tabela cliente com as regras: chave primária e preenchimento obrigatório.

```
create table cliente
(codigo_cliente number(4) constraint clie_cpf_pk primary key,
nome_cliente varchar2(20) constraint clie_nome_nn not null);
```

Sugestão para nome da constraint: clie – abreviação do nome da tabela, cpf – abreviação do nome da coluna, pk – abreviação de primary key.

Exemplo 1.1: (versão compacta)

```
create table cliente
(codigo_cliente number(4) primary key,
nome_cliente varchar2(20) not null);
```

Perceba que nesta versão não foi utilizada a palavra *constraint*, por isso não foi dado nome à regra, quem insere este nome é o SGBDR – Oracle, controlado pelo banco de dados, exemplo SYS_C0011212, no modelo anterior é possível personalizar este nome.

Exemplo 1.2: (nomeado as regras no final da instrução)

```
create table cliente
(codigo_cliente number(4) ,
nome_cliente varchar2(20) constraint clie_nome_nn not null,
constraint Clie_cpf_pk primary key (SQL codigo_cliente));
```

Nem todas as colunas precisam ter regras, algumas não têm e outras só possuem uma, outras colunas podem possuir mais de uma regra.

Criando a tabela vendedor, adicionando a regra de validação - *check*:

Tabela: Vendedor

Nome da coluna	Tipo de dados	Tamanho da coluna	Regra - <i>Constraint</i>
Codigo_vendedor	Numérico	4	Chave primária

Nome_vendedor	Alfanumérico	20	Não nulo
Salario_vendedor	Numérico	5,2	
Sexo_vendedor	Caracter	2	M ou F

```
create table vendedor
(codigo_vendedor number(4) constraint ven_cod_pk primary key,
nome_vendedor varchar2(20) constraint ven_nome_nn not null,
salario_vendedor number(7,2),
sexo char(1) constraint ven_sexo_ck check (sexo in
('M','F')));
```

Cuidado com o tipo de dados numéricos com casas decimais, pois não é possível armazenar vírgula. Há uma marcação para reconhecer a quantidade de inteiros e decimais.

No modelo físico da tabela vendedor, a coluna salário está com 5 inteiros e 2 decimais, para realizar esta criação em SQL deve-se somar a quantidade de inteiros e decimais e criar com o resultado da soma para os inteiros e deixar como está os decimais.

Na coluna sexo existe a regra de validação, ou seja, só é permitido cadastrar nesta coluna os valores M ou F.

Outra versão:

```
create table vendedor
(codigo_vendedor number(4) primary key,
nome_vendedor varchar2(20) not null,
salario_vendedor number(7,2),
sexo char(1) check (sexo in ('M','F')));
```

Criando a tabela produto: adicionando a regra de valores únicos e adicionando duas regras em uma única coluna.

Tabela: Produto

Nome da coluna	Tipo de dados	Tamanho da coluna	Regra - <i>Constraint</i>
Codigo_produto	Numérico	4	Chave primária
Nome_produto	Alfanumérico	20	Não nulo, valor único
preco_produto	Numérico	5,2	

```
create table produto
(codigo_produto number(4) constraint pro_cod_pk primary key,
nome_produto varchar2(20) constraint pro_nome_nn not null
constraint pro_nome_uk unique,
preco_produto number(7,2));
```

Ou

```
create table produto  
(codigo_produto number(4) primary key,  
nome_produto varchar2(20) not null unique,  
preco_produto number(7,2));
```

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

Interface SQL Plus, tabelas e regras

Biblioteca

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

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

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

Mapa do Site



Índice

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

Exemplificando a criação da estrutura de tabelas, regras e eliminação da estrutura

© Todos os direitos reservados

Ajuda?

PRÓXIMO

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

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura

da estrutura