

[< VOLTAR](#)

# Alteração de estrutura de uma tabela

Utilizar comando de alteração de estrutura, para demonstrar que, depois de pronta, a estrutura poderá ser modificada.

## NESTE TÓPICO

[> Alterando a estrutura de uma tabela](#)[> Referências](#)[Marcar tópico](#)

## Alterando a estrutura de uma tabela

Após a criação de uma estrutura de armazenamento, é possível efetuar alterações sem ter que eliminar a tabela e criá-la novamente. Para isso, usa-se o comando de alteração, que permite modificar partes da tabela que foram criadas de maneira errônea. Este comando também faz parte do subconjunto DDL.

Suponha que a "tabela cliente", que já fora criada anteriormente, não tenha uma coluna, contudo, a partir do uso do comando, é possível adicionar nova coluna após a criação da tabela. É importante salientar que as alterações que realizaremos em todas as tabelas não possuem dados cadastrados, portanto será possível efetuar tais alterações.

Sintaxe do comando de alteração:

1. `Alter table nome_tabela tipo_alteração nome_coluna tipo_dados(tamanho) [constraint nome_constraint tipo_constraint (nome_coluna)];`

Tipos de alteração:

- Add – adiciona nova coluna

- Modify ? modifica coluna existente
- Add constraint ? adiciona uma regra em uma coluna já existente
- Drop column ? elimina uma coluna existente
- Drop constraint ? elimina uma constraint existente
- Rename column ? renomeia a coluna

Criando uma tabela:

#### Tabela Cliente

Nome da coluna	Tipo de dados	Tamanho da coluna	Regra - Constraint
codigo_cliente	Numérico	4	Chave primária
Nome_cliente	Alfanumérico	20	Não nulo

1. Create table cliente
2. (codigo\_cliente number(4) constraint cli\_cod\_pk primary key,
3. Nome\_cliente varchar2(20) constraint cli\_nome\_nn not null);



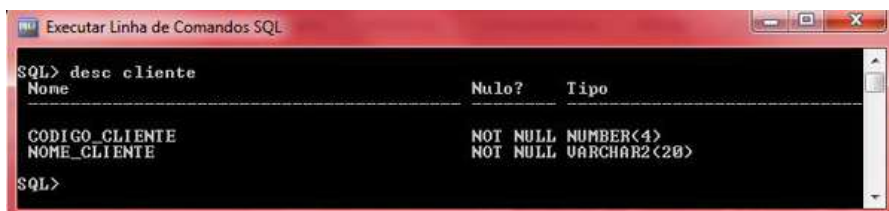
```

SQL> Create table cliente
2 <codigo_cliente number(4) constraint cli_cod_pk primary key,
3 Nome_cliente varchar2(20) constraint cli_nome_nn not null);

Tabela criada.
SQL> _
  
```

Visualizando a estrutura da tabela criada:

1. Desc cliente



```

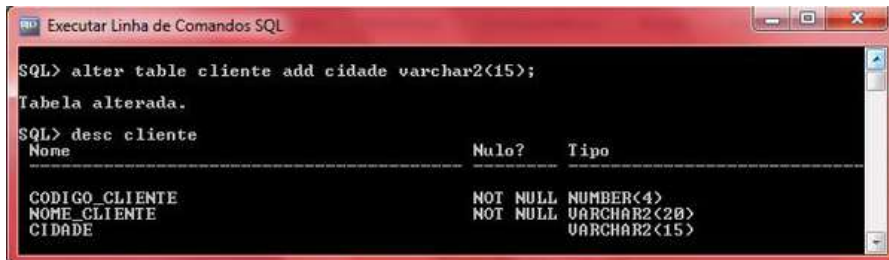
SQL> desc cliente
Nome
-----
CODIGO_CLIENTE          NOT NULL  NUMBER(4)
NOME_CLIENTE            NOT NULL  VARCHAR2(20)
SQL>
  
```

Exemplos:

Adicionando uma nova coluna sem regra na "tabela cliente" que já está criada:

1. `alter table cliente add cidade varchar2(15);`

Visualizando a estrutura da tabela alterada, adicionando uma coluna sem regra.

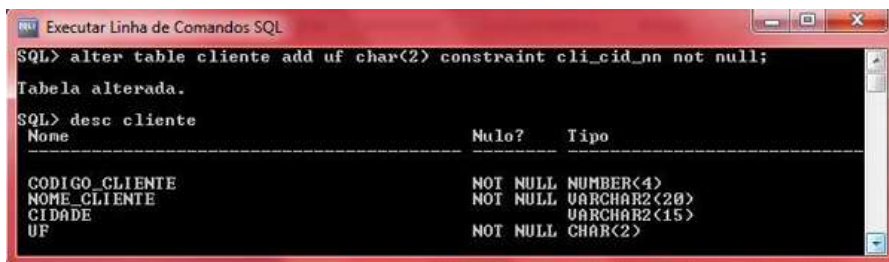


Agora temos, na "tabela cliente", a coluna "cidade", de tipo alfanumérica e com até 15 dígitos sem regra.

Adicionando uma nova coluna com regra e nome da regra na "tabela cliente" que já está criada:

1. `alter table cliente add uf char(2) constraint cli_cid_nn not null;`

Visualizando a estrutura da tabela criada, adicionando a coluna com sua regra e nome.



Adicionando uma coluna com regra, mas sem nome na regra na "tabela cliente" que já foi criada:

1. `alter table cliente add cpf_cliente char(11) unique;`

Adicionando uma regra em uma coluna já existente na tabela cliente que já foi criada:

1. `alter table cliente add constraint cli_cod_pk primary key (codigo_cliente);`

Adicionando uma regra do tipo chave estrangeira em uma coluna já existente na "tabela produto" que já está criada:

1. `alter table produto add constraint pro_cd_fk foreign key (codigo_tipo)`
2. `references tipo_produto;`

Existe uma exceção para adicionar a regra "não nulo" através do comando de alteração, como foi visto acima a variação do comando seria a opção "add constraint", mas para a adição desta regra utiliza-se a opção "modify", como abaixo:

Adicionando uma regra do tipo não nulo em uma coluna já existente na tabela produto que já está criada:

1. `alter table produto modify nome_produto not null;`

Modificando o tipo de dados de uma coluna já existente:

1. `alter table cliente modify cpf_cliente number(11);`

Modificando o tamanho de uma coluna já existente:

1. `alter table cliente modify cpf_cliente number(5);`

Modificando o tipo de dados e o tamanho simultaneamente de uma coluna já existente:

1. `alter table cliente modify cpf_cliente char(11);`

Eliminando apenas a regra existente em uma coluna:

1. `alter table cliente drop constraint cli_cod_pk;`

Para conseguir eliminar uma regra, é necessário saber o nome dela. Eliminando uma coluna existente na tabela:

1. `alter table cliente drop column cidade;`

Para renomear uma coluna:

1. `alter table cliente rename column nome_cliente to nome;`

## 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

Trabalhando a estrutura das tabelas, regras e relacionamento	Biblioteca ( <a href="https://www.unipr.br/biblioteca">https://www.unipr.br/biblioteca</a> )
--	---

as tabelas,

([https://www.uninove.br/conheca-](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 - pa

© Todos os direitos reservados

Ajuda?

PRÓXIMO:  
(https://ava.un  
nseridocso=)