

ALTER TABLE

Name

ALTER TABLE -- altera a definição da tabela

Synopsis

```
ALTER TABLE [ ONLY ] tabela [ * ]  
    ADD [ COLUMN ] coluna tipo [ restrição_de_coluna [ ... ] ]  
ALTER TABLE [ ONLY ] tabela [ * ]  
    ALTER [ COLUMN ] coluna { SET DEFAULT valor | DROP DEFAULT }  
ALTER TABLE [ ONLY ] tabela [ * ]  
    ALTER [ COLUMN ] coluna SET STATISTICS inteiro  
ALTER TABLE [ ONLY ] tabela [ * ]  
    RENAME [ COLUMN ] coluna TO novo_nome_da_coluna  
ALTER TABLE tabela  
    RENAME TO novo_nome_da_tabela  
ALTER TABLE tabela  
    ADD definição_de_restrição_de_tabela  
ALTER TABLE [ ONLY ] tabela  
    DROP CONSTRAINT restrição { RESTRICT | CASCADE }  
ALTER TABLE tabela  
    OWNER TO novo_dono
```

Entradas

tabela

O nome da tabela existente a ser alterada.

coluna

O nome de uma coluna nova ou existente.

tipo

O tipo da nova coluna.

novo_nome_da_coluna

O novo nome para a coluna existente.

novo_nome_da_tabela

O novo nome para a tabela.

definição_de_restrição_de_tabela

A nova restrição de tabela (table constraint) para a tabela.

novo_dono

O nome de usuário do novo dono da tabela.

Saídas

ALTER

Mensagem retornada se o nome da coluna ou da tabela for alterado com sucesso.

ERROR

Mensagem retornada se a tabela ou a coluna não existir.

Descrição

O comando `ALTER TABLE` altera a definição de uma tabela existente. A forma `ADD COLUMN` adiciona uma nova coluna na tabela utilizando a mesma sintaxe de [CREATE TABLE](#). A forma `ALTER COLUMN SET/DROP DEFAULT` permite definir ou remover o valor padrão para a coluna. Note que o valor padrão somente se aplica aos próximos comandos `INSERT`; as linhas existentes na tabela não são modificadas. A forma `ALTER COLUMN SET STATISTICS` permite controlar a coleta de estatísticas para as operações [ANALYZE](#) posteriores. A cláusula `RENAME` faz com que o nome da tabela, coluna, índice ou sequência seja mudado sem que os dados sejam modificados. Os dados permanecem do mesmo tipo e tamanho após o comando ser executado. A cláusula `ADD definição_de_restrição_de_tabela` adiciona uma nova restrição de tabela utilizando a mesma sintaxe de [CREATE TABLE](#). A cláusula `DROP CONSTRAINT restrição` elimina todas as restrições da tabela (e de suas descendentes) que correspondam à *restrição*. A cláusula `OWNER` muda o dono da tabela para o usuário *novo_dono*.

Somente o dono da tabela pode modificar seu esquema.

Notas

A palavra chave `COLUMN` é informativa podendo ser omitida.

Na atual implementação de `ADD COLUMN`, as cláusulas valor padrão e `NOT NULL` não são suportadas para a nova coluna. Pode ser usada a forma `SET DEFAULT` do comando `ALTER TABLE` para definir o valor padrão mais tarde. (Os valores atuais das linhas existentes poderão ser atualizados, posteriormente, para o novo valor padrão usando o comando [UPDATE](#).)

Em `DROP CONSTRAINT` a palavra chave `RESTRICT` é requerida, embora as dependências ainda não sejam verificadas. A opção `CASCADE` não é suportada. Atualmente `DROP CONSTRAINT` remove somente as restrições `CHECK`. Para remover as restrições `PRIMARY` ou `UNIQUE` deve ser removido o índice correspondente utilizando o comando [DROP INDEX](#). Para remover uma `FOREIGN KEY` é necessário recriar e recarregar a tabela usando outros parâmetros no comando [CREATE TABLE](#).

Por exemplo, para remover todas as restrições da tabela `distribuidores`:

```
CREATE TABLE temp AS SELECT * FROM distribuidores;
DROP TABLE distribuidores;
CREATE TABLE distribuidores AS SELECT * FROM temp;
DROP TABLE temp;
```

Somente o dono da tabela pode alterá-la. Mudar qualquer parte do esquema do catálogo do sistema não é permitido. O manual *Guia do Usuário do PostgreSQL* possui mais informações sobre herança.

Consulte o comando `CREATE TABLE` para obter uma descrição mais detalhada dos argumentos válidos.

Utilização

Para adicionar uma coluna do tipo `VARCHAR` à tabela:

```
ALTER TABLE distribuidores ADD COLUMN endereco VARCHAR(30);
```

Para mudar o nome de uma coluna existente:

```
ALTER TABLE distribuidores RENAME COLUMN endereco TO cidade;
```

Para mudar o nome de uma tabela existente:

```
ALTER TABLE distribuidores RENAME TO fornecedores;
```

Para adicionar uma restrição de verificação (`CHECK`) a uma tabela:

```
ALTER TABLE distribuidores ADD CONSTRAINT cep_chk CHECK (char_length(cod_cep) = 8);
```

Para remover uma restrição de verificação de uma tabela e de todas as suas filhas:

```
ALTER TABLE distribuidores DROP CONSTRAINT cepchk RESTRICT;
```

Para adicionar uma chave estrangeira a uma tabela:

```
ALTER TABLE distribuidores ADD CONSTRAINT fk_dist FOREIGN KEY (endereco) REFERENCES enderecos(endereco) MATCH FULL;
```

Para adicionar uma restrição de unicidade (multi-coluna) à tabela:

```
ALTER TABLE distribuidores ADD CONSTRAINT dist_id_cep_key UNIQUE (dist_id, cep);
```

Para adicionar uma restrição de chave primária a uma tabela com o nome gerado automaticamente, observando-se que a tabela somente pode possuir uma única chave primária:

```
ALTER TABLE distribuidores ADD PRIMARY KEY (dist_id);
```

Compatibilidade

SQL92

A forma `ADD COLUMN` está em conformidade, a não ser por não suportar valor padrão e `NOT NULL`, conforme foi explicado anteriormente. A forma `ALTER COLUMN` está em conformidade total.

O SQL92 especifica algumas funcionalidades adicionais para o comando `ALTER TABLE` que ainda não são diretamente suportadas pelo PostgreSQL:

```
ALTER TABLE tabela DROP [ COLUMN ] coluna { RESTRICT | CASCADE }
```

Remove a coluna da tabela. Na implementação atual, para remover uma coluna existente a tabela deve ser recriada e recarregada:

```
CREATE TABLE temp AS SELECT did, cidade FROM distribuidores;
DROP TABLE distribuidores;
CREATE TABLE distribuidores (
    did          DECIMAL(3)  DEFAULT 1,
    cidade       VARCHAR(40) NOT NULL
);
INSERT INTO distribuidores SELECT * FROM temp;
DROP TABLE temp;
```

As cláusulas para mudar o nome das tabelas, colunas, índices e seqüências são extensões do PostgreSQL ao SQL92.

[Prev](#)
ALTER GROUP

[Home](#)
[Up](#)

[Next](#)
ALTER USER