

Restrições em colunas

O tipo de dado é uma forma de limitar os dados que podem ser armazenados na tabela. Entretanto, para muitas aplicações, esta restrição ainda é abrangente demais. Por exemplo: uma coluna contendo preços de produtos normalmente só poderia aceitar valores positivos, mas não existe nenhum tipo de dado que aceite apenas números positivos. Outro problema é que pode ser necessário restringir os dados de uma coluna em relação a outras colunas ou linhas. Por exemplo, em uma tabela contendo informações relativas aos produtos deve haver apenas uma linha contendo um determinado código de produto.

Para esta finalidade, a linguagem SQL permite definir restrições em colunas e tabelas. As restrições permitem um nível de controle mais apurado e granular sobre os dados de uma tabela. Se o usuário tentar armazenar dados em uma coluna violando a restrição, o MySQL irá lançar um erro e irá desfazer a operação que originou a manipulação do dado que infringiu a restrição.

Restrição NOT NULL


Antes de saber o que é uma restrição **NOT NULL**, é necessário entender o que é o valor **NULL** para o MySQL.

No MySQL o valor **NULL** indica “desconhecido” ou “não aplicável”. É importante lembrar que o valor **NULL** é diferente de espaço, vazio e falso – essa é uma confusão relativamente comum.

Uma restrição **NOT NULL** simplesmente especifica que uma coluna não pode conter o valor nulo.

Um exemplo da sintaxe:

```
1
2 CREATE TABLE produtos (
3     codigo integer NOT NULL,
4     nome     text     NOT NULL,
5     preco    numeric
6 );
7
```

MySQL

Na tabela acima, ao inserirmos um produto nesta tabela, os campos *código* e *nome* devem ser preenchidos obrigatoriamente (pois eles estão com a restrição **NOT NULL**), enquanto o *preço* pode ser não especificado. Ao criar uma coluna e não especificar a restrição **NOT NULL**, a coluna, por padrão, aceitará valores nulos.

Restrições em colunas


Restrição UNIQUE

Existem situações onde é necessário que os valores de uma coluna sejam diferentes para cada registro... Em uma tabela que armazena produtos, por exemplo, cada produto deveria possuir um código único. Por isso, a coluna “código” não deveria conter valores repetidos para mais de um registro.

Pensando na solução do problema acima, o MySQL implementa a restrição **UNIQUE**. Ao inserir ou atualizar um valor em uma coluna que contenha essa restrição, o MySQL analisa todos os outros valores existentes nessa coluna e, se já existir esse valor para alguma das tuplas, um erro será apresentado.


Por exemplo:

```
1
2 CREATE TABLE produtos (
3     codigo integer UNIQUE,
4     nome     text,
5     preco    numeric
6 );
7
```




Acima, a restrição está sendo definida como restrição de coluna. Mas é possível defini-la como restrição de tabela:

```
1
2 CREATE TABLE produtos (
3     codigo integer,
4     nome     text,
5     preco    numeric,
6     UNIQUE (codigo)
7 );
8
```



A vantagem da definição com restrição de tabela é que, com ela, é possível atribuir um nome à restrição de unicidade:


```
1
2 CREATE TABLE produtos (
3     codigo integer,
4     nome     text,
5     preco    numeric,
6     CONSTRAINT nome_da_restricao UNIQUE(codigo)
7 );
8
```



Outra vantagem é que é possível definir um grupo de colunas:

Restrições em colunas

```
1  
2 CREATE TABLE exemplo (  
3   a integer,  
4   b integer,  
5   c integer,  
6   UNIQUE (a, c)  
7 );  
8
```

MySQL

Assim, a restrição de unicidade é violada quando existirem pelo menos duas linhas na tabela em que os valores, de cada uma das colunas correspondentes e que fazem parte da restrição, são iguais.