

Os usuários SQL são frequentemente confrontados com valores NULL em suas consultas e precisam processá-los corretamente. A função COALESCE() ajuda a lidar com os valores NULL.

As tabelas SQL armazenam dados em registros, e os registros são compostos de campos. Pode haver situações em que não sabemos o valor para um campo específico. Por exemplo, vamos supor que temos uma tabela com dados para `persons`. Tem os campos `first_name`, `last_name`, e `marital_status`. Quando não conhecemos o `marital_status` para uma determinada pessoa, SQL nos permite atribuir um valor NULL para este campo. Entretanto, o valor NULL não significa que a pessoa não tem um `marital_status`; significa apenas "não conhecemos esse valor".

Em outras palavras, SQL usa valores NULL para representar a ausência de valor. Entretanto, os valores NULL podem ser complicados de manusear.

O que faz a COALESCE()?

Em bancos de dados SQL, qualquer tipo de dado admite NULL como um valor válido; ou seja, qualquer coluna pode ter um valor NULL, independentemente do tipo de dado que seja. (Obviamente, algumas colunas serão obrigatórias (não-nuláveis), mas isto é definido pelo projetista do banco de dados, não o tipo de dados em si). Vamos mostrar um exemplo simples usando a tabela `persons`.

first_name	last_name	marital_status
Charles	Leclerc	single
Fernando	Alonso	married
George	Graue	NULL

Podemos usar a função SQL `COALESCE()` para substituir o valor NULL por um texto simples:

Code

```
SELECT
  first_name,
  last_name,
  COALESCE(marital_status, 'Unknown')
FROM persons
```

Na consulta acima, a função `COALESCE()` é utilizada para retornar o valor 'Unknown' somente quando `marital_status` é NULL. Quando `marital_status` não é NULL, `COALESCE()` retorna o valor da coluna `marital_status`. Em outras palavras, `COALESCE()` retorna o primeiro argumento não NULL.

Conheça os dados do exemplo

Para o restante do artigo, demonstraremos a função `COALESCE()` utilizando a tabela `stock` que é mostrado abaixo.

product	brand	subcategory	category	family	units	quantity_available	minimum_to_have
pork ribs	NULL	pork meat	meat	food	Kilos	400	130
tomatoes	Mr Red	NULL	vegetables	food	Kilos	280	100
lettuce	NULL	Leaf vegetables	NULL	food	Kilos	280	125
bananas	Big Brasil	NULL	vegetables	food	Kilos	450	150
hamburger	MaxBurg	cow meat	meat	food	Box	245	100
hamburger	RoyalBurg	cow meat	meat	food	Box	125	NULL
hamburger	SuperBurga	NULL	NULL	NULL	Box	200	80

Esta tabela armazena registros de dados do produto para um mercado e inclui as colunas `product`, `brand`, `subcategory`, `category`, `family`, `units`, `quantity_available` (o estoque atual deste produto), e `minimum_to_have` (o limiar quando o mercado precisa encomendar este produto de seus fornecedores).

Você notará que alguns dos produtos têm uma subcategoria, mas outros não. Por exemplo, o produto "costelas de porco" pertence à subcategoria "carne de porco" na categoria "carne" e a família "alimento". O produto "tomate" pertence à categoria "vegetais" e a família "alimento"; não tem uma subcategoria, portanto existe um NULL neste campo.

Exemplo de Consultas SQL usando a função COALESCE()

Agora, vamos ver como usar a função `COALESCE()` em alguns exemplos realistas.

Exemplo 1: Use COALESCE() para Substituir NULL por um Rótulo

Queremos mostrar todos os produtos com sua subcategoria, categoria e família. Entretanto, há alguns produtos com um NULL em sua categoria ou subcategoria. Para estes produtos, queremos exibir um texto: 'No Category' ou 'No Subcategory'. Aqui está a consulta que usaríamos:

Code

```
SELECT product,  
       COALESCE(subcategory, 'No Subcategory') AS subcategory,  
       COALESCE(category, 'No Category') AS category,  
       COALESCE(family, 'No Family') AS family  
FROM stock
```

Estamos usando a função `COALESCE()` para substituir os valores NULL por um texto. Você pode ver o resultado abaixo:

product	subcategory	category	family
pork ribs	pork meat	meat	food
tomatoes	No Subcategory	vegetables	food
lettuce	Leaf vegetables	No Category	food
bananas	No Subcategory	vegetables	food
hamburger	cow meat	meat	food
hamburger	cow meat	meat	food
hamburger	No Subcategory	No Category	No Family

Exemplo 2: Use COALESCE() para substituir NULL por um Valor Calculado

A função SQL `COALESCE()` também pode ser usada para calcular ou estimar um valor quando este valor não está presente. Por exemplo, cada produto tem um limite (representado pela coluna `minimum_to_have`) que requer um novo pedido ao fornecedor. Entretanto, alguns registros poderiam ter um valor NULL na coluna `minimum_to_have`; neste caso, podemos definir que o limite será 50% da coluna `quantity_available`. A consulta para calcular a estimativa do valor limite é:

Code

```
SELECT product,  
       quantity_available,  
       minimum_to_have,  
       COALESCE(minimum_to_have, quantity_available * 0.5) AS threshold  
FROM   stock
```

A função `COALESCE()` aqui retorna `minimum_to_have` quando o valor `minimum_to_have` não é NULL. Se `minimum_to_have` for NULL, então `COALESCE()` retornará `quantity_available * 0.5`

product	quantity_available	minimum_to_have	threshold
pork ribs	400	130	130
tomatoes	280	NULL	140
lettuce	280	125	125
bananas	450	150	150
hamburger	245	100	100
hamburger	125	100	100
hamburger	200	80	80

Vantagem de COALESCE() Para processar valores NULL

Existe várias maneiras de usar a função SQL `COALESCE()`. Cobrimos como usar `COALESCE()` para substituir valores NULL, como calcular um valor alternativo, entre outros exemplos.