

COMANDO CASE

Objetivos:

Este roteiro tem como objetivo explicar a utilização do comando CASE para aprimorar SELECTs.

Introdução

Quando existe a necessidade de verificarmos certas condições em uma atualização de informações de tabelas surge a necessidade de expressões que trabalhem com condições de análise no banco de dados. A expressão CASE tem a capacidade de combinar comandos UPDATE alterando informações mediante condições especificadas.

Para o exemplo deste código, utilizaremos um banco de dados de uma locadora, onde existe a necessidade de avaliar no banco de dados atual como podemos classificar os filmes que estão inseridos de forma incorreta. Atualmente os filmes possuem valores de TRUE e FALSE para determinar se o filme é de comédia, terror, ação, etc.

Abra seu editor preferido e desenvolva um script para a criação do banco de dados chamado LOCADORA.sql

```
mysql> desc tabela_filme;
```

Field	Type	Null	Key	Default	Extra
id_filme	int(11)	NO	PRI	NULL	auto_increment
titulo	varchar(50)	NO		NULL	
censura	varchar(5)	NO		NULL	
drama	varchar(10)	NO		NULL	
comedia	varchar(10)	NO		NULL	
acao	varchar(10)	NO		NULL	
horror	varchar(10)	NO		NULL	
ficcao	varchar(10)	NO		NULL	
infantil	varchar(10)	NO		NULL	
desenho	varchar(10)	NO		NULL	
data_compra	date	NO		NULL	

11 rows in set (0.02 sec)

```
mysql> SELECT titulo FROM tabela_filme;
```

titulo
Big Advenure
Shiny Things, The
End of the Line
A Rat named Darcy
Paraskavedekatriaphobia
Mad Clowns
Greg: The Untold Story
Potentially Habitable Planet
Angry Pirate
Shark Bait
Take it Back
Todo Poderoso
Senhor dos Anéis
Se eu fosse Você 2

14 rows in set (0.00 sec)

Detalhes do Banco de Dados Locadora:

Neste banco de dados, os filmes estão sendo organizados com colunas onde foram criados campos específicos para cada categoria de filme, cada filme que pertencer a determinada categoria é definido com um T (true) ou F (false) para informar que o filme pertence ou não àquela categoria. Veja abaixo uma parte da visualização de nossa tabela de filmes:

```
mysql> select titulo,censura,drama,comedia,acao from tabela_filme;
```

titulo	censura	drama	comedia	acao
Big Advenure	0	F	F	F
Shiny Things, The	13	T	F	F
End of the Line	18	T	F	F
A Rat named Darcy	0	F	F	F
Paraskavedekatriaphobia	18	T	T	T
Mad Clowns	18	F	F	F
Greg: The Untold Story	13	F	F	T
Potentially Habitable Planet	13	F	T	F
Angry Pirate	13	F	T	F
Shark Bait	0	F	F	F
Take it Back	18	F	T	F
Todo Poderoso	22	F	T	T
Senhor dos Anéis	22	F	F	T
Se eu fosse Você 2	22	T	T	T

14 rows in set (0.00 sec)

Percebemos que na situação atual é bastante complicado de identificar a qual categoria o filme pertence. Caso quiséssemos elaborar uma regra para classificar as categorias possíveis dos filmes, precisaríamos criar um campo que receberia unicamente a categoria definida e poderíamos utilizar o que já foi estudado anteriormente para tal classificação.

Uma forma de realizar a tarefa seria:

- Criar um novo campo para receber diretamente os nomes de categorias;
- Criar uma regra para a classificação;

A regra de classificação poderia ser definida da seguinte maneira:

- Se coluna drama = T então categoria = 'drama';
- Se coluna comedia = T então categoria = 'comédia';
- Se coluna horror = T então categoria = 'terror';
- Se coluna ficcao = T então categoria = 'Ficção';
- Se coluna desenho = T e Censura = 0 então categoria = 'família';
- Se coluna desenho = T e Censura <> 0 então categoria = 'diversos';

Para fazermos estas alterações, o que conhecemos atualmente seria através de vários UPDATES tabela com o uso de SET utilizando cláusulas where e operadores de comparação.

Exemplo;

```
UPDATE tabela_filme SET categoria = 'drama' WHERE drama = 'T';
UPDATE tabela_filme SET categoria = 'comedia' WHERE comedia = 'T';
UPDATE tabela_filme SET categoria = 'horror' WHERE horror = 'T';
UPDATE tabela_filme SET categoria = 'ficcao' WHERE ficcao = 'T';
```

Isto seria um bom exemplo de como poderíamos proceder, porém tivermos tabelas com uma infinidade de registros ficaria realmente bastante complicado, como alternativa para a redução de trabalho temos um outro comando que pode auxiliar nestas situações. O comando CASE poderá nos ajudar a realizar esta tarefa.

Sintaxe do comando CASE:

```
SET nova_coluna =
CASE
    WHEN coluna1 = valor1
        THEN novovalor1
    WHEN coluna2 = valor2
        THEN novovalor2
ELSE novovalor3
END;
```

Na situação da tabela de exemplo o comando poderá resolver o problema da classificação, vejamos o exemplo a seguir:



EXEMPLO DA UTILIZAÇÃO DO CASE:

EXEMPLO 1: classificando os filmes existentes e jogando o resultado para uma nova coluna denominada categoria.

- Criamos a nova coluna, chamada aqui de **CATEGORIA**:

```
mysql> alter table tabela_filme
-> add column categoria varchar(15);
Query OK, 14 rows affected (0.08 sec)
Records: 14 Duplicates: 0 Warnings: 0
```

- Verificamos o resultado com o comando DESC tabela_filme;

```
mysql> desc tabela_filme;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_filme | int(11) | NO | PRI | NULL | auto_increment |
| titulo | varchar(50) | NO | | NULL | |
| censura | varchar(5) | NO | | NULL | |
| drama | varchar(10) | NO | | NULL | |
| comedia | varchar(10) | NO | | NULL | |
| acao | varchar(10) | NO | | NULL | |
| horror | varchar(10) | NO | | NULL | |
| ficcao | varchar(10) | NO | | NULL | |
| infantil | varchar(10) | NO | | NULL | |
| desenho | varchar(10) | NO | | NULL | |
| data_compra | date | NO | | NULL | |
| categoria | varchar(15) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.03 sec)
```

- **APLICANDO O COMANDO CASE:** Aplicamos o comando para alterar a tabela com o uso do **CASE**:
 - O comando visualizado:

```
UPDATE tabela_filme
SET categoria =
CASE
    WHEN drama = 'T' THEN 'drama'
    WHEN comedia = 'T' THEN 'comedia'
    WHEN acao = 'T' THEN 'acao'
    WHEN horror = 'T' THEN 'terror'
    WHEN ficcao = 'T' THEN 'ficcao'
    WHEN infantil = 'T' THEN 'familia'
    WHEN desenho = 'T' THEN 'familia'
ELSE
    'diversos'
END;
```

Note que estamos validando para cada ocorrência de um T (true) de valor informado na coluna;

- O comando sendo aplicado:

```
mysql> UPDATE tabela_filme
-> SET categoria =
-> CASE
-> WHEN drama = 'T' THEN 'drama'
-> WHEN comedia = 'T' THEN 'comedia'
-> WHEN acao = 'T' THEN 'acao'
-> WHEN horror = 'T' THEN 'terror'
-> WHEN ficcao = 'T' THEN 'ficcao'
-> WHEN infantil = 'T' THEN 'familia'
-> WHEN desenho = 'T' THEN 'familia'
-> ELSE
-> 'diversos'
-> END;
Query OK, 14 rows affected (0.00 sec)
Rows matched: 14 Changed: 14 Warnings: 0
```



O resultado sendo visualizado:

```
mysql> SELECT titulo,categoria FROM tabela_filme;
+-----+-----+
| titulo                | categoria |
+-----+-----+
| Big Adventure         | família  |
| Shiny Things, The     | drama    |
| End of the Line       | drama    |
| A Rat named Darcy     | família  |
| Paraskavedekatriaphobia | drama    |
| Mad Clowns            | terror   |
| Greg: The Untold Story | acao     |
| Potentially Habitable Planet | comedia  |
| Angry Pirate          | comedia  |
| Shark Bait            | família  |
| Take it Back          | comedia  |
| Todo Poderoso         | comedia  |
| Senhor dos Anéis      | acao     |
| Se eu fosse Você 2    | comedia  |
+-----+-----+
14 rows in set (0.00 sec)
```

Detalhes Importantes:

- O comando **CASE** validará a **primeira** ocorrência encontrada e realizará a alteração solicitada.
- Isto significa que **no primeiro T encontrado** o restante do código será ignorado.
- Preste atenção nos filmes que possuem mais de uma coluna com T selecionado, como o caso do filme **Paraskavedekatriaphobia**, que foi classificado como **DRAMA**.
- Podemos concluir que a ordem informada no comando CASE importa muito para que as alterações possam ser validadas com sucesso.

OBSERVAÇÕES:

- É obrigatório o uso do ELSE? Se não quisermos uma alternativa para quando nenhuma condição for satisfeita não necessitamos o uso dele, pode ser omitido;
- O CASE pode ser utilizado também com as instruções SELECT, INSERT, DELETE e como utilizamos agora, o UPDATE;



INCREMENTANDO O CASE

Podemos adicionar algumas funcionalidades no CASE desenvolvido incluindo cláusulas AND e OR nas condições elaboradas.

Por exemplo, se desejássemos que o case também validasse os filmes na categoria família para aceitar apenas filmes que tenham censura = 0, podemos incluir uma pequena alteração no comando, segue abaixo o exemplo.

EXEMPLO 1: adicionando o AND combinado com CASE:

Aplicando o comando:

```
UPDATE tabela_filme
SET categoria =
CASE
    WHEN drama = 'T' THEN 'drama'
    WHEN comedia = 'T' THEN 'comedia'
    WHEN acao = 'T' THEN 'acao'
    WHEN horror = 'T' THEN 'terror'
    WHEN ficcao = 'T' THEN 'ficcao'
    WHEN infantil = 'T' THEN 'familia'
    WHEN desenho = 'T' AND censura = 0 THEN 'familia'
ELSE
    'diversos'
END;
```

```
mysql> UPDATE tabela_filme
-> SET categoria =
-> CASE
-> WHEN drama = 'T' THEN 'drama'
-> WHEN comedia = 'T' THEN 'comedia'
-> WHEN acao = 'T' THEN 'acao'
-> WHEN horror = 'T' THEN 'terror'
-> WHEN ficcao = 'T' THEN 'ficcao'
-> WHEN infantil = 'T' THEN 'familia'
-> WHEN desenho = 'T' AND censura = 0 THEN 'familia'
-> ELSE
-> 'diversos'
-> END;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 14 Changed: 0 Warnings: 0
```

Agora somente filmes que tiverem censura = 0 serão considerados como da categoria Família.



EXEMPLO 2: alterando as categorias para adicionar a validação de censura nos filmes cadastrados.

Para testarmos novamente as funcionalidades do CASE, vamos fazer algumas modificações:

- Agora validaremos as seguintes categorias na classificação verificando a censura:
 - terror ≥ 18
 - acao ≥ 15
 - drama ≥ 13
 - comedia ≥ 10
 - ficção ≥ 10
 - E qualquer filme que estiver na categoria diversos e possuir a censura = 0 deverá ser classificado como família.
- Caso contrário o filme será considerado da categoria **diversos**;

Tente executar este exemplo para ver como ficará o seu comando CASE:

Vejamos inicialmente como estão os filmes atualmente: Execute um select para selecionar os campos título, censura e categoria da tabela_filme.

título	censura	categoria
Big Adventure	0	família
Shiny Things, The	13	drama
End of the Line	18	drama
A Rat named Darcy	0	família
Paraskavedekatriaphobia	18	drama
Mad Clowns	18	terror
Greg: The Untold Story	13	acao
Potentially Habitable Planet	13	comedia
Angry Pirate	13	comedia
Shark Bait	0	família
Take it Back	18	comedia
Todo Poderoso	22	comedia
Senhor dos Anéis	22	acao
Se eu fosse Você 2	22	comedia

14 rows in set (0.00 sec)

Temos de criar uma regra para que possa adequar a todas os critérios informados anteriormente validando a categoria a ser selecionada juntamente com o nível de censura do filme em questão.



Detalhes para montar o CASE:

- Precisamos verificar a qual categoria o filme está atualmente pertencendo;
- Precisamos encontrar uma maneira de validar a censura com operadores de comparação;
- Precisamos adequar a situação da categoria diversos e censura 0 para que vá para a categoria família;
- Precisamos adequar o comando para que quando nenhuma condição for satisfatória o filme seja encaminhado para a categoria diversos;

APLICANDO O COMANDO:

```
UPDATE tabela_filme
SET categoria =
CASE
    WHEN drama = 'T' AND censura >=13 THEN 'drama'
    WHEN comedia = 'T' AND censura >=10 THEN 'comedia'
    WHEN acao = 'T' AND censura >=15 THEN 'acao'
    WHEN horror = 'T' AND censura >= 18 THEN 'terror'
    WHEN ficcao = 'T' AND censura >= 10 THEN 'ficcao'
    WHEN infantil = 'T' AND censura = 0 THEN 'família'
    WHEN categoria = 'diversos' AND censura = 0 THEN 'família'
ELSE
    'diversos'
END;
```

Verificando o resultado:

```
mysql> SELECT titulo,censura,categoria from tabela_filme;
```

titulo	censura	categoria
Big Advenure	0	família
Shiny Things, The	13	drama
End of the Line	18	drama
A Rat named Darcy	0	família
Paraskavedekatriaphobia	18	drama
Mad Clowns	18	terror
Greg: The Untold Story	13	diversos
Potentially Habitable Planet	13	comedia
Angry Pirate	13	comedia
Shark Bait	0	família
Take it Back	18	comedia
Todo Poderoso	22	comedia
Senhor dos Anéis	22	acao
Se eu fosse Você 2	22	comedia

14 rows in set (0.00 sec)

Note que todos os filmes agora foram classificados obedecendo a nossas regras.

Por fim, vamos deletar todas as colunas que não são mais necessárias ao banco de dados:



```
mysql> ALTER TABLE tabela_filme
-> DROP COLUMN drama,
-> DROP COLUMN comedia,
-> DROP COLUMN acao,
-> DROP COLUMN horror,
-> DROP COLUMN ficcao,
-> DROP COLUMN infantil,
-> DROP COLUMN desenho;
Query OK, 14 rows affected (0.13 sec)
Records: 14 Duplicates: 0 Warnings: 0
```

E verificamos o resultado final:

```
mysql> DESC tabela_filme;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_filme | int(11) | NO | PRI | NULL | auto_increment |
| titulo | varchar(50) | NO | | NULL | |
| censura | varchar(5) | NO | | NULL | |
| data_compra | date | NO | | NULL | |
| categoria | varchar(15) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```