

 05

## Filtrando usando maior, menor e diferente

### Transcrição

Você pode fazer o [download \(\[https://caelum-online-public.s3.amazonaws.com/1220-mysqlintroducaoaosql/05/SQL\\\_12.sql\]\(https://caelum-online-public.s3.amazonaws.com/1220-mysqlintroducaoaosql/05/SQL\_12.sql\)\)](https://caelum-online-public.s3.amazonaws.com/1220-mysqlintroducaoaosql/05/SQL_12.sql) completo do código realizado neste vídeo e continuar seus estudos.

[00:00] Vamos melhorar um pouco os filtros, para buscarmos, por exemplo, coisas maiores ou menores a alguma referência.

[00:08] Eu vou criar um novo script e vamos dar um SELECT \* FROM tbCliente. Temos, por exemplo, o campo idade. Se eu fizer isso: SELECT \* FROM tbCliente WHERE IDADE = 22;

[00:47] Lembrando que o número inteiro não precisa de aspas. Vou rodar esse comando e note que tenho quatro clientes com mais de 22 anos. Mas posso também selecionar os clientes que têm mais de 22 anos: SELECT \* FROM tbCliente WHERE IDADE > 22;

[01:33] Assim, vou ver todo mundo que tem mais de 22 anos. E se eu quiser ver os que tem menos de 22 anos: SELECT \* FROM tbCliente WHERE IDADE < 22;

[02:02] São três clientes.

[02:08] Agora, eu gostaria de ver os que têm menos de 22 anos, inclusive quem tem 22 anos. Então, ao invés de menor, coloco menor ou igual: SELECT \* FROM tbCliente WHERE IDADE <= 22;

[02:23] O igual vai fazer com que o 22 entre no critério do filtro.

[02:40] E se eu quiser selecionar todo mundo, exceto os que têm 22 anos: SELECT \* FROM tbCliente WHERE IDADE <> 22;

[02:55] Esse é o símbolo de diferente. Se eu olhar, não tenho ninguém com 22 anos.

[03:08] Agora, uma pergunta que alguém pode estar pensando: posso aplicar o menor, maior, menor igual, maior igual a textos? Qual vai ser o comportamento dele?

[03:26] Tenho os nomes dos meus clientes. Eu posso fazer o seguinte: SELECT \* FROM tbCliente WHERE NOME > 'Fernando Cavalcante';

[03:50] Ele listou assim alguns clientes. Para o MySQL, existe internamente também uma ordem, não só dos números, mas das letras também, a ordem alfabética. Então, para o MySQL, o B é maior que A. O C é maior que B. O X é maior que R. Respeitando uma ordem alfabética.

[04:29] Quando ele acha dois nomes que começam com a mesma letra, ele usa como critério a segunda letra. Se a segunda letra for igual, a terceira. E assim por diante.

[04:43] Nesse caso, como é maior, usei como critério o Fernando, ou seja, a letra F. Todo mundo que foi listado na condição são pessoas que começam com o nome acima de F.

[05:02] Fátima não entraria, porque como o nome Fernando é Fe, se eu tivesse uma cliente chamada Fátima, o F empataria com o F de Fernando, então poderia ser um candidato. Mas ao comparar a letra E com a letra A de Fátima, Fátima ficaria

antes. Então não entraria na condição.

[05:33] Claro que se eu colocar maior ou igual, o próprio Fernando entra no filtro.

[05:45] Também posso usar o diferente. Vai vir todo mundo, mas não o Fernando.

[06:08] Eu queria chamar a atenção para um detalhe que é motivo de erros no MYSQL. Quando nós fazemos uma consulta, achamos que está certo, vemos um erro e não conseguimos descobrir o motivo.

[06:27] Eu agora vou fazer um SELECT \* FROM tbProduto. Tenho meu preço de lista. Ele não é um inteiro. É um FLOAT. Vamos fazer isso: SELECT \* FROM tbProduto WHERE PRECO\_LISTA = 16.008

[07:10] Eu tenho esse cara na lista. Tento executar essa condição, mas ele não me traz nada. Por quê? Internamente, o FLOAT é um ponto flutuante. E como ponto flutuante, ele não consegue achar exatamente o resultado que quero ver.

[07:46] Ou seja, se eu quisesse trabalhar com condições iguais, seria mais interessante usar o tipo decimal, porque aí sim pegaria exato. Valores com ponto flutuante são difíceis de criar condições usando o igual. Podemos usar o maior, ele vai me mostrar todo mundo com preço maior. Posso usar o menor também. Mas o maior igual, menor igual ou igual não vou conseguir usar. E nem o diferente, porque ele também não vai conseguir achar o cara exatamente.

[08:54] Se eu usar o diferente, vou conseguir rodar, mas o cara com preço 16.008 também vai aparecer.

[09:07] Existe um comando mais complexo de where chamado between. Se eu quisesse realmente pegar quem fosse 16.008 eu teria que fazer isso: SELECT \* FROM tbProduto WHERE PRECO\_LISTA BETWEEN 16.008 AND 16.009;

[09:52] Eu peço que ele liste todos os produtos cujo preço está entre 16.007 e 16.009. se eu rodar o comando, ele me lista o 16.008. Mas essa é uma característica do MYSQL. Como é FLOAT, não consigo exatamente obter uma condição que seja ele especificamente. Mas podemos pegar limites inferiores e superiores bem pequenos para poder achar o cara propriamente dito.