Nesse vídeo, estudaremos **funções**. São nelas que residem as maiores diferenças entre os vários bancos que utilizam SQL. Neste curso, estamos conhecendo essa linguagem usando o **MySQL**. A maior parte do que aprendemos até agora (como DISTINCT e JOIN) é muito parecido ou igual entre o MySQL, o Oracle ou o SQL Server, por exemplo, citando os três principais bancos relacionais do mercado.

Ao tratar de funções, no entanto, nos deparamos com algumas diferenças — como você utiliza uma função em um banco nem sempre é igual à do outro banco. Quando temos um projeto desenvolvido em SQL para MySQL e desejamos migrá-lo para Oracle ou SQL Server, uma ação muito importante é verificar o uso das funções e fazer a conversão pontual delas.

A seguir, veremos alguns exemplos de funções e vamos classificá-las em três grandes grupos: **funções escalares**, que utilizam textos; **funções de data** que, como o próprio nome sugere, manipulam datas; e **funções matemáticas**, que realizam operações entre campos *float* ou inteiros na nossa base de dados.

É humanamente impossível saber de cabeça todas as funções que o MySQL disponibiliza, mas por sorte temos a internet para nos ajudar. Um dos recursos que podemos acessar é o [site da documentação oficial do MySQL](https://dev.mysql.com/doc/). Essa documentação é muito bem detalhada e aborda tudo o que precisamos saber sobre MySQL. Nesse site, podemos clicar em "*MySQL Reference Manual*", selecionar "*MySQL 8.0 Reference Manual*" e seremos redirecionados para a [página do Manual de Referência do MySQL 8.0](https://dev.mysql.com/doc/refman/8.0/en/), que é a versão que estamos usando. À esquerda, veremos o índice de capítulos, dentre eles haverá um chamado "*[Functions and Operators](https://dev.mysql.com/doc/refman/8.0/en/functions.html" \t "_blank)*", no qual encontraremos uma lista de funções que podemos utilizar no banco de dados.

<https://dev.mysql.com/doc/>

<https://dev.mysql.com/doc/refman/8.0/en/>

<https://dev.mysql.com/doc/refman/8.0/en/functions.html>

Outro site interessante é o [w3school.com](https://www.w3schools.com/), que também nos oferece uma gama de materiais específicos sobre bancos de dados relacionais, inclusive o MySQL, que é o nosso foco. Nesse vídeo, para abordar as funções de texto, vamos usar esse site como apoio, pois ele é mais simples e fácil de visualizar do que a documentação oficial.

Então, começaremos na [página de funções do MySQL](https://www.w3schools.com/sql/sql_ref_mysql.asp). À esquerda, abaixo do título "*String Functions*", temos uma lista enorme de funções de textos. A seguir, vamos revisar algumas das mais importantes.

No vídeo, ao navegar pelos sites, optamos pela tradução automática do inglês para o português para tornar o conteúdo mais acessível. Nesse processo, algumas funções também tiveram seus nomes traduzidos. Para que elas funcionem, precisamos usar o nome original, em inglês.

A [função CONCAT()](https://www.w3schools.com/sql/func_mysql_concat.asp) concatena *strings*. Ou seja, aplicando CONCAT() sobre duas *strings*, é possível uni-las em uma única sequência de texto. Podemos juntar um número ilimitado de *strings*, separando-as com vírgula:



A [função LTRIM()](https://www.w3schools.com/sql/func_mysql_ltrim.asp)tira espaços à esquerda (*left*) de um texto. *Trim*, em português, significa "aparar" e é exatamente isso que a função faz:



A [função RTRIM()](https://www.w3schools.com/sql/func_mysql_rtrim.asp), por outro lado, vai remover os espaços à direita (*right*) da *string*:

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Já se o nosso objetivo for aparar o texto dos dois lados, podemos usar a [função TRIM()](https://www.w3schools.com/sql/func_mysql_trim.asp), que retira espaços tanto da direita quanto da esquerda. Os espaços no meio da *string* não são removidos:



Outro exemplo interessante é a [função LCASE()](https://www.w3schools.com/sql/func_mysql_lcase.asp), que transforma um texto que está em caixa-alta (ou seja, em maiúsculas) deixando-o com letras minúsculas (em inglês, *lowercase*):



Uma alternativa é o [LOWER()](https://www.w3schools.com/sql/func_mysql_lower.asp), que faz o mesmo processo:



Já a [função UCASE()](https://www.w3schools.com/sql/func_mysql_ucase.asp) fará o contrário, deixando toda a *string* em maiúsculas (em inglês, *uppercase*):



Um equivalente a essa função é o [UPPER()](https://www.w3schools.com/sql/func_mysql_upper.asp):



A [função SUBSTRING()](https://www.w3schools.com/sql/func_mysql_substring.asp) também é bastante usada. Com ela, conseguimos retirar um pedaço de texto de dentro de uma *string* maior. Para usá-la, devemos informar três parâmetros: a *string* original, a posição onde começaremos a retirada do texto e o número de caracteres que vamos extrair a partir da posição que especificamos:

Tela preta com letras brancas

O conteúdo gerado por IA pode estar incorreto.

Nesse caso, o texto original é "**SQL Tutorial**", vamos começar a extrair a partir do **quinto elemento** (a letra T, maiúscula) e pegaremos **três caracteres**. Então, a *subtring* resultante será "Tut".

A [função LENGTH()](https://www.w3schools.com/sql/func_mysql_length.asp) retorna o tamanho de uma *string*:

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Nesse caso, o retorno será 12, pois o espaço também é contado.

Existe uma série de funções úteis e você pode continuar explorando o site para conhecê-las um pouco melhor. Não vamos nos aprofundar neste vídeo, mas você pode, por exemplo, ler mais sobre a [função INSTR()](https://www.w3schools.com/sql/func_mysql_instr.asp), que faz uma pesquisa para saber se uma *string* está dentro de outra *string*.

Dependendo dos nossos objetivos, podemos sempre consultar a documentação do MySQL ou fóruns na internet para buscar uma função específica que resolva o problema que estamos enfrentando em determinado projeto.

Agora, vamos ao MySQL Workbench criar alguns exemplos com funções. Abriremos um novo script e começaremos com a seguinte consulta:



Note que fizemos um SELECT sem o FROM, sem selecionar uma tabela. É como se fôssemos exibir uma constante. Para nomear a coluna, usaremos a expressão AS:



Dessa forma, aplicamos LTRIM() sobre uma *string* que contém vários espaços e o resultado será o texto "OLÁ", com os espaços **à esquerda** aparados. Podemos fazer algo semelhante com RTRIM():

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Nesse caso, aparamos os espaços **à direita**. Se nossa meta for remover espaços de ambos os lados de um texto, manipularemos o texto com a função TRIM():

Tela de celular

O conteúdo gerado por IA pode estar incorreto.

O MySQL Workbench dará sugestões de preenchimento automático à medida que digitamos, dando alguns palpites de funções.

A seguir, vamos concatenar *strings*:

Texto

O conteúdo gerado por IA pode estar incorreto.

Note que um dos parâmetros é um **espaço**. Sem ele, as palavras "OLÁ" e "TUDO" apareceriam juntas.

Também é possível fazer testes com as funções de texto que transformam letras minúsculas em maiúsculas:



E vice-versa:



Para extrair uma sequência de *strings* dentro de um texto maior, optamos pela função SUBSTRING():



Perceba que informamos apenas dois parâmetros. O primeiro é o texto original e o segundo é a posição onde começamos a extrair a *substring* - a sexta posição, ocupada pela letra T. Na falta do terceiro parâmetro que especifique quantos caracteres retirar, o MySQL continuará até o fim do texto. Assim, o resultado dessa consulta será "TUDO BEM?".

Caso seja esclarecida a quantidade de caracteres desejada, então o retorno será diferente:



Nesse caso, a *substring* começa no T e, contando quatro caracteres a partir dessa letra, termina na letra O. O resultado será a palavra "TUDO".

Esses foram alguns exemplos livres utilizando dados bem simples para embasar nossos conhecimentos de funções que manipulam textos. Em um cenário real, usaremos essas funções dentro de um SELECT com informações presentes em tabelas, por exemplo:



Veja que, dessa vez, selecionamos uma tabela com o FROM. Para cada linha do nosso SELECT, a função CONCAT() foi aplicada com campos diferentes da tabela — o nome e o CPF entre parênteses de cada cliente.