Um processo bastante recorrente quando trabalhamos com SQL é a **conversão de dados**. Cada campo das tabelas tem um tipo que condiz com seu conteúdo e que não deve mudar, porém nem sempre esse tipo será apropriado para certas manipulações (como algumas funções). Às vezes, precisamos exibir datas em um formato específico, convertidas em *string*, ou usar um dado de tipo numérico como texto, por exemplo. Não convém modificar o tipo da coluna na base de dados, visto que foi criada assim para melhor conveniência. A solução é a conversão de dados no momento em que formos usá-los — tal como em uma função.

Vamos começar aprendendo sobre conversão de datas em *strings*, que é um processo extenso, com inúmeros parâmetros e que merece uma explicação mais detalhada. Como apoio, usaremos o **Manual de Referências** da documentação oficial do MySQL, mais especificamente [o subcapítulo "*Date and Time Functions*", com ênfase na função DATE\_FORMAT()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_date-format).

Essa é uma função específica para conversão do tipo data para *strings*. O primeiro parâmetro dela é a data, enquanto o segundo parâmetro indica o formato — este é o ponto mais trabalhoso.

No Manual, na descrição da função DATE\_FORMAT(), há uma tabela com uma série de **especificadores** que se iniciam com o símbolo de porcentagem (%). Cada um deles é responsável por um tipo diferente de conversão e se relaciona com um segmento em particular da data (segundo, minuto, hora etc.). Para ficar mais claro, vamos criar alguns testes no MySQL Workbench.

Abriremos um novo script e começaremos com uma consulta simples, selecionando a data atual no formato *timestamp*:



O retorno será um dado no formato de data. A título de exemplo, vamos tentar rodar uma função de texto com esse dado em formato de data:



A função CONCAT() teoricamente manipula somente textos. Ao executar a consulta acima, no entanto, ela consegue trabalhar com um dado em formato de data, pois se trata de uma **conversão implícita**. Vale ressaltar, no entanto, que nem todas as funções são capazes de realizar essa conversão automaticamente.

Note que essa data é exibida no formato padrão (ano, mês, dia, horas, minutos e segundos). Agora, com a função DATE\_FORMAT() e os **especificadores** que vimos há pouco, podemos modificar essa visualização. Vamos começar usando %Y:

Relógio digital com números

O conteúdo gerado por IA pode estar incorreto.

Nosso primeiro parâmetro é a data atual. Já o segundo parâmetro diz respeito ao formato: na tabela que consultamos, no Manual de Referências, a descrição de %Y indica a conversão do ano com **quatro** dígitos. Podemos optar por outro formato, por exemplo, usando o %y (com "y" minúsculo) e a exibição do ano terá apenas **dois** dígitos:

Relógio digital com números

O conteúdo gerado por IA pode estar incorreto.

Ademais, é possível concatenar com outros **especificadores** e até símbolos, como a barra:

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

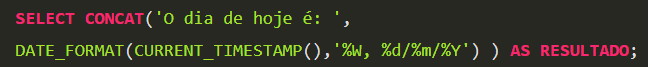
Agora, teremos mês e ano. No caso, %m exibirá o número do mês com dois dígitos. Se nosso objetivo for apresentar o mês com apenas um dígito, nós vamos à tabela e procuramos como fazer essa mudança — com %c.

Um padrão bastante usado é o dia, seguido do mês (com dois dígitos) e o ano (com quatro dígitos):

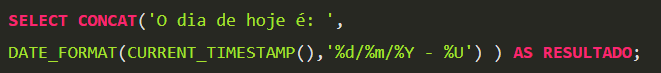
Relógio digital com números

O conteúdo gerado por IA pode estar incorreto.

Ademais, podemos acrescentar o dia da semana (com %W):



Também temos a opção de adicionar o número da semana (com %U). Ou seja, se o resultado mostrar o número 15, quer dizer que estamos na 15ª semana do ano:



Note que, além das barras que separam a data, também podemos usar vírgulas e traços.

É possível fazer diversas combinações a depender do formato de exibição de nossa preferência. Você pode continuar testando esses formatos no seu projeto, consultando a tabela do Manual de Referências.

Finalmente, aprenderemos sobre a função CONVERT(), que faz a conversão de um tipo para outro (especificado como segundo parâmetro). Por exemplo, vamos transformar um valor numérico em char:

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

No retorno, é difícil distinguir o **tipo** desse registro. Para confirmar a conversão, podemos executar uma função que só admite textos, como o SUBSTRING():



Dessa vez, o retorno será 2, o que quer dizer que a função SUBSTRING() identificou "23.3" como um texto e fez a extração da *substring* (um caractere a partir do primeiro).

Funções e conversões são tópicos extremamente amplos, neste curso não convém nos prolongar em explicações detalhadas sobre esses assuntos. Passamos por alguns exemplos e agora você já pode prosseguir com seus estudos, consultando a documentação e buscando em fóruns online quando necessário.