



01000101 01110011 01100011 01101111 01101100 01100001 00100000  
01100100 01100101 00100000 01000100 01100001 01100100  
01101111 01110011 00001010

# GLOSSÁRIO SQL

SQL: FUNÇÕES DE STRING

// Glossário SQL\_

# FUNÇÕES DE STRING



Essas funções realizam uma variedade de operações, como ajustar a capitalização com **UPPER** e **LOWER**, extrair subconjuntos de caracteres com **SUBSTR** ou **SUBSTRING**, remover espaços indesejados com **TRIM**, e substituir partes de uma string com **REPLACE**. Elas também permitem concatenar strings com **CONCAT** ou operadores como **||**.

---

Por exemplo, nessa consulta utilizamos a função **LENGTH** para conferir se todas as linhas de registros da coluna **CPF** da tabela **Colaboradores** contenham 11 caracteres, ou seja, tenham um número de CPF preenchido sem faltar nenhum dígito:

```
SELECT COUNT(*), LENGTH(cpf) qtd  
FROM Colaboradores  
WHERE qtd = 11;
```

Nessa consulta, criamos uma frase usando o operador **||** para concatenar textos com informações contidas em algumas colunas da tabela **Colaboradores**. Além de aplicar a função **UPPER** para trazer todo o texto com caracteres maiúsculos.

```
SELECT UPPER
```

```
('A pessoa colaboradora ' || nome || ' de CPF '  
|| cpf || ' possui o seguinte endereço: ' ||  
endereco) as texto  
FROM Colaboradores;
```

Esta consulta é útil para analisar os prefixos dos endereços dos **clientes**. Utilizando **SUBSTRING**, extraímos os primeiros cinco caracteres do campo endereço, o que pode ajudar na categorização rápida de áreas geográficas ou na validação de dados de endereço:

```
SELECT nome, SUBSTRING(endereco, 1, 5) AS  
EnderecoInicio  
FROM Clientes;
```

Dados muitas vezes são inseridos com espaços extras que podem causar erros em operações de busca e verificação. A função **TRIM** remove espaços indesejados antes e depois do texto do email, garantindo que operações subsequentes com esses dados sejam precisas e sem falhas.

```
SELECT nome, TRIM(email) AS EmailLimp  
FROM Clientes;
```

Modificar termos nos dados para atender a novos padrões de nomenclatura é uma necessidade comum. **REPLACE** altera todos os endereços que começam com "Rua" para "Av.", facilitando uma padronização rápida e eficiente dos registros de endereço.

```
SELECT nome, REPLACE(endereco, 'Rua', 'Av.') AS  
EnderecoAtualizado  
FROM Clientes;
```



## UTILIZE E DOMINE O SQL!

Parabéns por explorar o **Glossário de SQL**! Agora que você adquiriu os comandos mais avançados da linguagem, é hora de aplicar esse conhecimento. Utilize este material como referência em seus projetos e desafios, praticando para aprimorar suas habilidades na manipulação de bancos de dados. Ao se tornar mais confiante na linguagem SQL, você estará preparado para **enfrentar novos desafios**.

Muito obrigado por chegar até aqui e nos vemos nos próximos cursos da formação em SQL da Alura. Até mais!

# AVALIE O CURSO E DEIXE UM COMENTÁRIO.

**Compartilhe** um resumo de seus novos conhecimentos em suas redes sociais.

alura



Escola Data Science