



## Faça como eu fiz

- 1) No console do **BigQuery** verifique se está com o usuário correto.
- 2) No exemplo para testar a função **LOWER** vá na janela do editor de consultas e digite:

```
WITH items AS
  (SELECT 'FOO' AS ITEM
   UNION ALL SELECT 'BAR'
   UNION ALL SELECT 'BAZ')
SELECT LOWER(ITEM) FROM items;
```

[COPIAR CÓDIGO](#)

- 3) Para testar a função **UPPER** copie:

```
WITH items AS
  (SELECT 'foo' AS ITEM
   UNION ALL SELECT 'bar'
   UNION ALL SELECT 'baz')
SELECT UPPER(ITEM) FROM items;
```

[COPIAR CÓDIGO](#)

- 4) Usando **LOWER** e **UPPER** use o código:

```
WITH items AS
  (SELECT 'rUA Antonio Carlos' AS ITEM
   UNION ALL SELECT 'AVENIDA SÃO PAULO'
```

```
UNION ALL SELECT 'tv atroS DO aLT0')  
SELECT ITEM, UPPER(ITEM), LOWER(ITEM) FROM items;
```

[COPIAR CÓDIGO](#)

5) A função **INITCAP** funciona com o código:

WITH examples AS

```
(SELECT "Alo Mundo-todo mundo!" AS FRASES  
UNION ALL SELECT "o cachorro TORNADO é alegre+manso"  
UNION ALL SELECT "maça&laranja&pera"  
UNION ALL SELECT "tata ta tavendo a tatia")  
SELECT FRASES, INITCAP(FRASES) FROM examples;
```

[COPIAR CÓDIGO](#)

6) Para usar delimitadores junto com o **INITCAP** copie o código:

WITH examples AS

```
(SELECT "Alo Mundo-todo mundo!" AS FRASES, " " AS DELIMITER  
UNION ALL SELECT "o cachorro TORNADO é alegre+manso", "+"  
UNION ALL SELECT "maça&laranja&pera", "&"  
UNION ALL SELECT "tata ta tavendo a tatia", "t")  
SELECT FRASES, INITCAP(FRASES), INITCAP(FRASES, DELIMITER) FROM
```

[COPIAR CÓDIGO](#)

7) Para mostrar o uso das funções **LTRIM**, **RTRIM** e **TRIM** use como exemplo o código:

WITH items AS

```
(SELECT "      MAÇA      " AS ITEM
```

```
UNION ALL SELECT "      BANANA      "  
UNION ALL SELECT "      LARANJA      "  
SELECT ITEM, LTRIM(ITEM), RTRIM(ITEM), TRIM(ITEM) FROM items;
```

[COPIAR CÓDIGO](#)

8) As funções **LEFT** e **RIGHT** também podem ser combinadas como neste código.

```
WITH items AS  
(SELECT "      MAÇA      " AS ITEM  
UNION ALL SELECT "      BANANA      "  
UNION ALL SELECT "      LARANJA      "  
SELECT TRIM(ITEM), LEFT(TRIM(ITEM), 2), RIGHT(TRIM(ITEM), 2) FROM
```

[COPIAR CÓDIGO](#)

9) Usando a função **CONCAT** para concatenar campos de uma tabela. Use o código:

```
WITH examples AS  
(SELECT "DR" AS Titulo, "Carlos" as NOME, "Junior" as SOBRENOME  
UNION ALL SELECT "SR", "Marcos", "Almeida"  
UNION ALL SELECT "DR" , "Mario", "Costa"  
UNION ALL SELECT "MS" , "Maria", "Rosa")  
SELECT CONCAT (Titulo, " ", Nome, " ", Sobrenome) FROM examples;
```

[COPIAR CÓDIGO](#)

10) Usando a função **CHAR\_LENGTH** para retornar o número de caracteres que o string possui. Digite o código:

WITH examples AS

```
(SELECT "DR" AS Titulo, "Carlos" as NOME, "Junior" as SOBRENOME
UNION ALL SELECT "SR", "Marcos", "Almeida"
UNION ALL SELECT "DR" , "Mario", "Costa"
UNION ALL SELECT "MS" , "Maria", "Rosa")
SELECT CONCAT (Titulo, " ", Nome, " ", Sobrenome),
CHAR_LENGTH(CONCAT (Titulo, " ", Nome, " ", Sobrenome)) FROM exar
```

COPIAR CÓDIGO

11) Usando a função **STARTS\_WITH** para os campos que iniciam com Dr e terminam com a letra a . Digite o código:

WITH examples AS

```
(SELECT "DR" AS Titulo, "Carlos" as NOME, "Junior" as SOBRENOME
UNION ALL SELECT "SR", "Marcos", "Almeida"
UNION ALL SELECT "DR" , "Mario", "Costa"
UNION ALL SELECT "MS" , "Maria", "Rosa")
SELECT CONCAT (Titulo, " ", Nome, " ", Sobrenome),
CHAR_LENGTH(CONCAT (Titulo, " ", Nome, " ", Sobrenome)),
STARTS_WITH(CONCAT (Titulo, " ", Nome, " ", Sobrenome), "DR"),
ENDS_WITH(CONCAT (Titulo, " ", Nome, " ", Sobrenome), "Dr") FROM
```

COPIAR CÓDIGO

12) Continuando com as funções de string temos a **INSTR** que funciona conforme o código:

WITH example AS

```
(SELECT 'banana' AS source_value, 'an' AS search_value, 1 as p
UNION ALL SELECT 'banana' AS source_value, 'an' AS search_value,
```

```
UNION ALL SELECT 'banana' AS source_value, 'xx' AS search_value,  
SELECT *, INSTR(source_value, search_value, position, occurrence
```

[COPIAR CÓDIGO](#)

13) A função **SUBSTR** neste exemplo retorna os três caracteres a partir da posição 3 como no código:

```
WITH example AS  
(SELECT 'banana' AS source_value,  
UNION ALL SELECT 'melancia'  
UNION ALL SELECT 'tangerina')  
SELECT source_value, SUBSTR(source_value,3,3) FROM example;
```

[COPIAR CÓDIGO](#)

14) A função **SUBSTR** neste exemplo retorna todos os caracteres a partir da posição 3 como no código:

```
WITH example AS  
(SELECT 'banana' AS source_value,  
UNION ALL SELECT 'melancia'  
UNION ALL SELECT 'tangerina')  
SELECT source_value, SUBSTR(source_value,3) FROM example;
```

[COPIAR CÓDIGO](#)

15) **STRPOS** neste exemplo retorna a posição do caractere @ :

```
WITH example AS  
(SELECT 'foo@example.com' AS source_value,
```

```
UNION ALL SELECT 'victor@gmail.com'  
UNION ALL SELECT 'quexample@brazil.com')  
SELECT source_value, STRPOS(source_value, "@") FROM example;
```

[COPIAR CÓDIGO](#)

16) Este código usa as funções **SUBSTR** e **STRPOS** combinadas.

```
WITH example AS  
(SELECT 'foo@example.com' AS source_value,  
UNION ALL SELECT 'victor@gmail.com'  
UNION ALL SELECT 'quexample@brazil.com')  
SELECT source_value, SUBSTR(source_value,1, STRPOS(source_value,
```

[COPIAR CÓDIGO](#)

17) Neste exemplo vemos o uso da função **REVERSE**

```
WITH example AS  
(SELECT 'foo@example.com' AS source_value,  
UNION ALL SELECT 'victor@gmail.com'  
UNION ALL SELECT 'quexample@brazil.com')  
SELECT source_value, REVERSE(source_value) FROM example;
```

[COPIAR CÓDIGO](#)

18) **REPLACE** é usado desta forma:

```
WITH example AS  
(SELECT 'foo@example.com' AS source_value,  
UNION ALL SELECT 'victor@gmail.com'
```

```
UNION ALL SELECT 'quexample@brazil.com')
SELECT source_value, REPLACE(source_value, "@", "XXXXXX") FROM exa
```

COPIAR CÓDIGO

19) A função **SPLIT** é mostrada no código:

```
WITH example AS
(SELECT 'foo@example.com' AS source_value,
UNION ALL SELECT 'victor@gmail.com'
UNION ALL SELECT 'quexample@brazil.com')
SELECT source_value, SPLIT(source_value, "@") FROM example;
```

COPIAR CÓDIGO

20) Uma **Expressão Regular** é uma representação para que você encontre padrões em um texto.

21) O exemplo mostra o uso das funções **REGEXP\_CONTAINS**, **REGEXP\_EXTRACT**, **REGEXP\_EXTRACT\_ALL** e **REGEXP\_REPLACE**:

```
SELECT FIELD,
REGEXP_CONTAINS(FIELD, r'[0-9]{5}-[0-9]{3}') AS TEM_CEP,
REGEXP_EXTRACT(FIELD, r'[0-9]{5}-[0-9]{3}', 1, 1) AS CEP,
REGEXP_EXTRACT(FIELD, r'[0-9]{5}-[0-9]{3}', 1, 2) AS CEP2,
REGEXP_EXTRACT_ALL(FIELD, r'[0-9]{5}-[0-9]{3}') AS CEP3,
REGEXP_REPLACE(FIELD, r'[0-9]{5}-[0-9]{3}', 'XXXXX-XXX') AS CEP2,
FROM
(SELECT * from UNNEST
(["22222-22", "22222-222", "Meu CEP é 222222-22", "Do CEP 22222-22"]
```

COPIAR CÓDIGO

22) Para testar a função **FORMAT** use o código `SELECT FORMAT("%d", 10);`

23) O comando `SELECT FORMAT("%015d", 10);` retorna `000000000000010` como resposta.

24) Salve as consultas desta aula.