



Faça como eu fiz

- 1) No console do **BigQuery** verifique se está com o usuário correto.
- 2) No exemplo de operadores lógicos vá na janela do editor de consultas e digite:

```
SELECT
  gender, tripduration
FROM
  `bigquery-public-data`.new_york_citibike.citibike_trips
WHERE (tripduration < 600 AND gender = 'female') OR gender = 'r'
LIMIT 100;
```

[COPIAR CÓDIGO](#)

- 3) Para verificar o funcionamento do código faça os testes conforme o vídeo.
- 4) Nos códigos seguinte são testados os operadores lógicos:

```
WITH example AS (
  SELECT true AS is_valid, 'a' as letter, 1 as position
  UNION ALL SELECT false , 'b', 2
  UNION ALL SELECT false , 'c', 3
)
SELECT * FROM example;
```

[COPIAR CÓDIGO](#)

- 5) Teste os operadores lógicos usando o código:

```
WITH example AS (  
  SELECT true AS is_valid, 'a' as letter, 1 as position  
  UNION ALL SELECT false , 'b', 2  
  UNION ALL SELECT false , 'c', 3  
)  
SELECT * FROM example WHERE is_valid != false AND position > 0;
```

[COPIAR CÓDIGO](#)

6) Teste usando um campo lógico como condição:

```
WITH example AS (  
  SELECT true AS is_valid, 'a' as letter, 1 as position  
  UNION ALL SELECT false , 'b', 2  
  UNION ALL SELECT false , 'c', 3)  
SELECT * FROM example WHERE is_valid AND position > 0;
```

[COPIAR CÓDIGO](#)

7) Colocando um operador lógico para testar a cláusula WHERE alterando o código que acessa a base de dados pública:

```
SELECT  
  gender, tripduration  
FROM  
  `bigquery-public-data`.new_york_citibike.citibike_trips  
WHERE (tripduration < 600 AND gender = 'female')  
LIMIT 100;
```

[COPIAR CÓDIGO](#)

8) A expressão completa para a cláusula `WHERE` mostrando todos os operadores lógicos:

```
SELECT
  gender, tripduration
FROM
  `bigquery-public-data`.new_york_citibike.citibike_trips
WHERE (tripduration < 600 AND gender = 'female') = true
LIMIT 100;
```

[COPIAR CÓDIGO](#)

9) Para testar expressões condicionais use o código:

```
WITH catalog AS (
  SELECT 30.0 AS costPrice, 0.15 as margin, 0.1 as taxRate
  UNION ALL SELECT NULL, 0.21, 0.15
  UNION ALL SELECT 30.0, NULL, 0.09
  UNION ALL SELECT 30.0, 0.30, NULL
  UNION ALL SELECT 30.0, NULL, NULL
)
SELECT * FROM catalog;
```

[COPIAR CÓDIGO](#)

10) Após as alterações do vídeo o código fica assim:

```
WITH catalog AS (
  SELECT 30.0 AS costPrice, 0.15 as margin, 0.1 as taxRate
  UNION ALL SELECT NULL, 0.21, 0.15
  UNION ALL SELECT 30.0, NULL, 0.09
  UNION ALL SELECT 30.0, 0.30, NULL
  UNION ALL SELECT 30.0, NULL, NULL
)
```

```
SELECT ROUND (  
  IF (costPrice IS NULL, 30.0, costPrice) *  
  IF (margin IS NULL, 0.10, margin) *  
  IF (taxrate IS NULL, 0.15, taxrate) , 2  
) as FORMULA FROM catalog;
```

[COPIAR CÓDIGO](#)

11) No editor de consultas teste as expressões:

```
SELECT COALESCE ('A', 'B', 'C');
```

[COPIAR CÓDIGO](#)

E:

```
SELECT COALESCE (NULL, 'B', 'C');
```

[COPIAR CÓDIGO](#)

12) Aplicando **COALESE** na expressão anterior. Alteramos o código para:

```
WITH catalog AS (  
  SELECT 30.0 AS costPrice, 0.15 as margin, 0.1 as taxRate  
  UNION ALL SELECT NULL, 0.21, 0.15  
  UNION ALL SELECT 30.0, NULL, 0.09  
  UNION ALL SELECT 30.0, 0.30, NULL  
  UNION ALL SELECT 30.0, NULL, 0.10  
)  
SELECT  
  IF (costPrice IS NULL, 30.0, costPrice) *  
  IF (margin IS NULL, 0.10, margin) *  
  IF (taxrate IS NULL, 0.15, taxrate)  
  as FORMULA1 ,  
  COALESCE (
```

```
costPrice * margin * taxrate,  
30.0 * margin * taxrate,  
costprice * 0.10 * taxrate,  
costPrice * margin * 0.15  
) as FORMULA2 FROM catalog;
```

[COPIAR CÓDIGO](#)

13) Usamos o **SAFE_CAST** que é idêntico a **CAST** , exceto que retorna **NULL** em vez de gerar um erro:

```
WITH example AS (  
  SELECT 'Jonh' AS employee, 'Doente' as Hours_work  
  UNION ALL SELECT 'Jean', '100'  
  UNION ALL SELECT 'Peter', 'De férias'  
  UNION ALL SELECT 'Mary', '80'  
)  
SELECT SUM (SAFE_CAST(Hours_work AS INT64)) AS TOTAL FROM example
```

[COPIAR CÓDIGO](#)

14) O código seguinte mostra o uso da **CASE WHEN**:

```
WITH Numbers AS (  
  SELECT 90 as A, 2 as B  
  UNION ALL SELECT 50, 8  
  UNION ALL SELECT 60, 6  
  UNION ALL SELECT 50, 10  
)  
SELECT A, B,  
  CASE  
    WHEN (A = 90 AND B > 10) THEN 'red'  
    WHEN A = 50 THEN 'blue'
```

```
ELSE 'green' END AS Color  
FROM Numbers;
```

[COPIAR CÓDIGO](#)

15) Comparando o **CASE WHEN** e **RANGE_BUCKET** usando o código:

```
WITH Students AS  
(SELECT 'A1' AS ALUNO, 11 AS AGE  
UNION ALL SELECT 'A2' , 12  
UNION ALL SELECT 'A3' , 11  
UNION ALL SELECT 'A4' , 14  
UNION ALL SELECT 'A5' , 17  
UNION ALL SELECT 'A6' , 17  
UNION ALL SELECT 'A7' , 18  
UNION ALL SELECT 'A8' , 16  
UNION ALL SELECT 'A9' , 11  
UNION ALL SELECT 'A10' , 12  
UNION ALL SELECT 'A11' , 13  
UNION ALL SELECT 'A12' , 13  
UNION ALL SELECT 'A13' , 16)  
SELECT ALUNO, RANGE_BUCKET( AGE, [9, 13, 15, 18]),  
CASE  
  WHEN AGE >= 9 AND AGE < 13 THEN '1'  
  WHEN AGE >= 13 AND AGE < 15 THEN '2'  
  WHEN AGE >= 15 AND AGE < 18 THEN '3'  
  ELSE '4' END  
FROM Students;
```

[COPIAR CÓDIGO](#)

16) Salve as consultas desta aula.

