

Faça como eu fiz

1) No console do **BigQuery** verifique se está com o usuário correto.

2) Na janela de consultas digite:

```
WITH example AS
  (SELECT 'Sat' AS Day, 1451 AS numrides, 1018 AS oneways
  UNION ALL SELECT 'Sun', 2376, 936)
  SELECT *, (oneways/numrides) as frac_oneway,
  ROUND(oneways/numrides, 2) AS frac_noeway_round FROM example;
```

COPIAR CÓDIGO

3) Para verificar o funcionamento do **ROUND** faça os testes conforme o vídeo.

4) No código seguinte é prevenida a divisão por zero:

```
WITH example AS
  (SELECT 'Sat' AS Day, 1451 AS numrides, 1018 AS oneways
  UNION ALL SELECT 'Sun', 2376, 936
  UNION ALL SELECT 'Wed', 0, 0)
  SELECT *, ROUND(IEEE_Divide(oneways, numrides), 2) AS frac_noawa)
```

COPIAR CÓDIGO

5) A função **SAFE** retorna valor `NULL` caso haja um erro. Digite o código:

```
SELECT SAFE.LOG (10, -3), SAFE.LOG (10, 3);
```

[COPIAR CÓDIGO](#)

6) Alterando a consulta para a situação do vídeo com seguinte código:

```
WITH example AS
(SELECT 'Sat' AS Day, 1451 AS numrides, 1018 AS oneways
UNION ALL SELECT 'Sun', 2376, 936
UNION ALL SELECT 'Mon', NULL, NULL
UNION ALL SELECT 'Tue', IEEE_Divide(3,0), 0
UNION ALL SELECT 'Wed', IEEE_Divide(-3,0), 0
)
SELECT * FROM example
WHERE numrides < 2000;
```

[COPIAR CÓDIGO](#)

7) Para verificar a precisão dos cálculos numéricos copie o código:

```
WITH example AS
(SELECT 1.23 AS PAYMENT
UNION ALL SELECT 7.89
UNION ALL SELECT 12.43)
SELECT SUM(PAYMENT) AS TOTAL_PAYMENT, AVG(PAYMENT) AS AVG_PAYMENT
FROM example;
```

[COPIAR CÓDIGO](#)

Com resultado:

```
-- 21.55 (sum)
-- 7.1833333333 (avg)
```

[COPIAR CÓDIGO](#)

8) Para melhorar a precisão usamos o comando **NUMERIC** que converte um **FLOAT** em um **NUMERIC**:

```
WITH example AS
  (SELECT NUMERIC '1.23' AS PAYMENT
  UNION ALL SELECT NUMERIC '7.89'
  UNION ALL SELECT NUMERIC '12.43')
  SELECT SUM(PAYMENT) AS TOTAL_PAYMENT, AVG(PAYMENT) AS AVG_PAYMENT
  FROM example;
```

[COPIAR CÓDIGO](#)

9) Existem outras funções matemáticas que podem ser usadas. Estes são alguns exemplos que pode ser vistos no vídeo:

- **SIGN** - retorna um numero 1 ou 0. 1 se for positivo e 0 se for negativo

```
SELECT SIGN (-3.45);
```

[COPIAR CÓDIGO](#)

- **IS_INF** checa se o numero é infinito ou não

```
SELECT IS_INF(IEEE_DIVIDE(0,0)), IEEE_DIVIDE(0,0);
```

[COPIAR CÓDIGO](#)

- **IS_NAN** checa se o numero é NAN ou não

```
SELECT IS_NAN(IEEE_DIVIDE(3,0)), IEEE_DIVIDE(3,0);
```

[COPIAR CÓDIGO](#)

- RAND - Gera um número randomico. entre 0 e 1 (exceto o 1)

```
SELECT RAND();
```

[COPIAR CÓDIGO](#)

- SQRT - RAIZ QUADRADA

```
SELECT SQRT(144);
```

[COPIAR CÓDIGO](#)

- POW - Elevado a potencia

```
SELECT POW (2, 4);
```

[COPIAR CÓDIGO](#)

- LN, LOG, LOG10

```
SELECT LOG10(5);
```

[COPIAR CÓDIGO](#)

- GREATEST - Retorna o maior numero dentro de um array.

```
SELECT GREATEST (1,2,3,3,4,4,5,4,5,6,7,6,5,8,2,3,4);
```

[COPIAR CÓDIGO](#)

- LEAST - Retorna o maior numero dentro de um array.

```
SELECT LEAST (1,2,3,3,4,4,5,4,5,6,7,6,5,8,2,3,4);
```

[COPIAR CÓDIGO](#)

- EXPRESSÕES MATEMATICAS - SAFE_ADD, SAFE_SUBTRACT, SAFE_DIVIDE, SAFE_NEGATIVE

```
SELECT ((4+5)/3)*10;
```

```
SELECT SAFE_MULTIPLY (300000000000, 400000000000);
```

[COPIAR CÓDIGO](#)

- MOD - Resto

```
SELECT MOD (10,3);
```

[COPIAR CÓDIGO](#)

- ROUND, TRUNC

```
SELECT ROUND(3.42, 1), TRUNC(3.42, 1);
```

[COPIAR CÓDIGO](#)

- CEIL E FLOOR
 - CEIL MAIOR INTEIRO DEPOIS DO NUMERO
 - FLOOR MENOR INTEIRO ANTES DO NUMERO

```
SELECT CEIL(3.48), FLOOR(3.48);
```

[COPIAR CÓDIGO](#)

- SEN, COS, TAG, ACOS, ASEN, ATAG, SENH e outras.

10) Segundo o vídeo o uso do **RANGE_BUCKET**. O código seguinte responde à pergunta:

QUANTOS ALUNOS EU TENHO ENTRE 10 E 13, ENTRE 13 E 15 E ENTRE 15 E 18?

```
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 RANGE_BUCKET( AGE, [9, 13, 15, 19]), COUNT(*) FROM Students
  GROUP BY 1;
```

[COPIAR CÓDIGO](#)

