

 11

Faça como eu fiz

- 1) No console do BigQuery verifique se está com o usuário correto.
- 2) Veja a diferença dos resultados ao testar as funções de Data e Hora . Use o código:

```
SELECT CURRENT_DATETIME,  
CURRENT_TIMESTAMP,  
CURRENT_DATE,  
CURRENT_TIME;
```

[COPIAR CÓDIGO](#)

- 3) Podem ser incluídas informações geográficas como no exemplo:

```
SELECT CURRENT_DATETIME('America/Sao_Paulo'),  
CURRENT_DATETIME('Europe/London'),  
CURRENT_TIMESTAMP,  
CURRENT_DATE,  
CURRENT_TIME;
```

[COPIAR CÓDIGO](#)

- 4) Para definir Data e Hora use como exemplo:

```
SELECT TIMESTAMP('2020-07-01 10:00:00'),  
DATETIME(2020, 7, 1, 10, 0, 0),  
DATE(2020, 7, 1),  
TIME(10, 0, 0);
```

[COPIAR CÓDIGO](#)

5) **TIMESTAMP** pode ser usado como parâmetro em funções de Data e Hora .

```
SELECT DATE(TIMESTAMP('2020-07-01 10:00:00')),  
DATETIME(TIMESTAMP('2020-07-01 10:00:00')),  
TIME(TIMESTAMP('2020-07-01 10:00:00'));
```

[COPIAR CÓDIGO](#)

6) Operações matemáticas com datas. Verifique o que cada função retorna.

```
SELECT  
DATE_ADD (DATE(2008, 12, 25), INTERVAL 5 DAY) AS CINCO_DIAS_DEFINITIVO,  
DATE_ADD (DATE(2008, 12, 25), INTERVAL 4 YEAR) AS QUATRO_ANOS_DEFINITIVO,  
TIMESTAMP_ADD (CURRENT_TIMESTAMP, INTERVAL 45 MINUTE) AS QUAREMAMENTO;
```

[COPIAR CÓDIGO](#)

7) Para subtrair datas vemos o exemplo:

```
SELECT  
DATE_SUB (DATE(2008, 12, 25), INTERVAL 5 DAY) AS CINCO_DIAS_ANTES,  
DATE_SUB (DATE(2008, 12, 25), INTERVAL 4 YEAR) AS QUATRO_ANOS_ANTES,  
TIMESTAMP_SUB (CURRENT_TIMESTAMP, INTERVAL 45 MINUTE) AS QUAREMANTO_ANTES;
```

[COPIAR CÓDIGO](#)

8) Para saber os dias ou horas entre data usamos o exemplo:

```
SELECT  
DATE_DIFF (DATE(2010,12,25), DATE(2008, 9, 15), DAY),
```

```
DATETIME_DIFF (CURRENT_DATETIME, DATETIME(TIMESTAMP('2020-07-01')))
```

COPiar Código

- 9) Para extrair elementos de uma Data usamos **EXTRACT** como está mostrado a seguir:

```
SELECT DATA,  
       EXTRACT(MONTH FROM DATA) AS MES,  
       EXTRACT(DAY FROM DATA) AS DIA,  
       EXTRACT(YEAR FROM DATA) AS ANO,  
       EXTRACT(DAYOFWEEK FROM DATA) AS SEMANA  
FROM UNNEST (GENERATE_DATE_ARRAY('2015-12-23', '2016-01-09')) AS DATA  
ORDER BY DATA;
```

COPiar Código

- 10) Para testar e verificar as datas truncadas acompanhe no vídeo a explicação do código:

```
SELECT  
       DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY),  
       DATETIME_TRUNC(DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY)),  
       DATETIME_TRUNC(DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY)),  
       DATETIME_TRUNC(DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY)),  
       DATETIME_TRUNC(DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY))
```

COPiar Código

- 11) Os últimos dias do mês e ano podem ser obtidos com o código:

```
SELECT
```

```
DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY),  
LAST_DAY(DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY)), MONTH  
LAST_DAY(DATETIME_ADD(CURRENT_DATETIME, INTERVAL 90 DAY)), YEAR
```

[COPIAR CÓDIGO](#)

12) As funções baseadas em **FORMAT** (FORMAT_TIME, FORMAT_DATETIME, FORMAT_DATE, FORMAT_TIMESTAMP) usam a seguinte notação:

```
<FUNCAO> (<FORMATO>, VALOR)
```

[COPIAR CÓDIGO](#)

Onde podemos usar os seguintes elementos no campo **FORMATO**:

%A	O nome completo do dia da semana
%a	O nome abreviado do dia da semana
%B	O nome completo do mês
%b ou %h	O nome abreviado do mês
%C	O século (um ano dividido por 100 e truncado como um inteiro) como um número decimal
%D	A data no formato %m/%d/%y
%d	O dia do mês como número decimal (01-31)
%e	O dia do mês como número decimal (1-31). Dígitos únicos são precedidos por um espaço
%F	A data no formato %Y-%m-%d
%G	O ano ISO 8601 com o século como número decimal. Cada ano ISO começa na segunda-feira
%g	O ano ISO 8601 sem o século como número decimal (00-99). Cada ano ISO começa na segunda-feira
%j	O dia do ano como número decimal (001-366)
%m	O mês como número decimal (01-12)

%n	Um caractere de nova linha
%Q	O trimestre como um número decimal (de 1-4)
%t	Um caractere de tabulação
%U	O número da semana do ano (domingo como o primeiro dia da semana) como número decimal (0-7)
%u	O dia da semana (segunda-feira como o primeiro dia da semana) como número decimal (1-7)
%V	O número da semana ISO 8601 do ano (segunda-feira como o primeiro dia da semana) como número decimal (0-53)
%W	O número da semana do ano (segunda-feira como o primeiro dia da semana) como número decimal (0-53)
%w	O dia da semana (domingo como o primeiro dia da semana) como número decimal (0-6)
%x	A representação de data no formato MM/DD/YY
%Y	O ano com o século como número decimal
%y	O ano sem o século como número decimal (00-99), com um zero opcional à esquerda. Pode ser usado com %E4Y
%E4Y	Anos com quatro caracteres (0001 ... 9999). %Y produz quantos caracteres forem necessário
%H	A hora em um relógio de 24 horas como número decimal (00-23)
%I	A hora em um relógio de 12 horas como número decimal (01-12)
%j	O dia do ano como número decimal (001-366)
%k	A hora em um relógio de 24 horas como número decimal (0-23). Dígitos únicos são precedidos por zero
%l	A hora em um relógio de 12 horas como número decimal (1-12). Dígitos únicos são precedidos por zero
%M	O minuto como número decimal (00-59)
%n	Um caractere de nova linha
%P	am ou pm
%p	AM ou PM
%R	A hora no formato %H:%M
%r	A hora em um relógio de 12 horas usando a notação AM/PM
%S	O segundo como número decimal (00-60)
%T	A hora no formato %H:%M:%S
%t	Um caractere de tabulação
%X	A representação da hora no formato HH:MM:SS

%%	Um único caractere %
%E#S	Segundos com # dígitos de precisão fracionária
%E*S	Segundos com precisão fracionária total (um literal '*')

13) A expressão é exemplo para a função **FORMAT** com datas:

```
SELECT CURRENT_DATETIME, FORMAT_DATETIME('"%A, Dia %d de %B de %Y'
```

[COPIAR CÓDIGO](#)

14) Digite o código que acessa uma tabela pública para acompanhar o resultado no vídeo:

```
SELECT  
    visitStartTime FROM  
`bigquery-public-data.google_analytics_sample.ga_sessions_20170`  
LIMIT 10;
```

[COPIAR CÓDIGO](#)

15) Para retornar

```
SELECT  
    visitStartTime, TIMESTAMP_SECONDS(visitStartTime) FROM  
`bigquery-public-data.google_analytics_sample.ga_sessions_20170`  
LIMIT 10;
```

[COPIAR CÓDIGO](#)

16) O código retorna a quantidade de segundos até hoje desde 01-01-1970 .

```
SELECT CURRENT_TIMESTAMP, UNIX_SECONDS(CURRENT_TIMESTAMP);
```

COPiar CÓDIGO

17) Salve as consultas desta aula.