**Tipos de Conversão**

* Operador : :
* CAST
* Resumo
* O operador :: e o CAST() são funções utilizadas para converter um dado para a unidade desejada
* O operador :: é mais “Clean”, porém, em algumas ocasiões não funciona, sendo necessário utilizar a função CAST()

Exemplo 1 : Conversão de texto em data

– Corrija a query abaixo utilizando o operador : :

select ‘2021-10-01’::date - ‘2021-02-01’::date

ou

select nome\_coluna::date

from nome\_tabela

Exemplo 2: Conversão de texto em número

– Corrija a query abaixo utilizando o operador ::

select ‘100’ - ‘10’

select ‘100’::numeric - ‘10’::numeric

Exemplo 3: Conversão de número em texto

– Corrija a query abaixo utilizando o operador ::

select replace(112122, ‘1’, ‘A’)

select replace(112122::text, ‘1’, ‘A’)

Exemplo 4: Conversão de texto em data

– Corrija a query abaixo utilizando a função CAST

select cast(‘2021-10-01’ as date) - cast(‘2021-02-01’ as date)

* **TRATAMENTO GERAL**

Exemplo 1: Agrupamento de dados com CASE WHEN

– Calcule o n° de clientes que ganham abaixo de 5k, entre 5k e 10k, entre 10k e 15k e acima de 15k

with faixa\_de\_renda as (

select

income,

case

when income < 5000 then ‘0-5000’

when income >= 5000 and income < 10000 then ‘5000-10000’

when income >= 10000 and income < 15000 then ‘10000-15000’

else ‘15000+’

end as faixa\_renda

from sales.customers

)

select faixa\_renda, count(\*)

from faixa\_de\_renda

group by faixa\_renda

Exemplo 2: Tratamento de dados nulos com COALESCE

– Crie uma coluna chamada populacao\_ajustada na tabela temp\_tables.regions e preencha com os dados da coluna population, mas caso esse campo estiver nulo, preencha com a população média (geral) das cidades do Brasil

select

\*,

case

when population is not null then population

else (select avg(population) from temp\_tables.regions)

end as populacao\_ajustada

from temp\_tables.regions

where population is null

**ou utilizando o coalesce**

select

\*,

coalesce(population, (select avg(population) from temp\_tables.regions)) as populacao\_ajustada

from temp\_tables.regions

**Resumo**

* CASE WHEN é o comando utilizado para criar respostas específicas para diferentes condições e é muito utilizado para fazer agrupamento de dados
* COALESCE é o comando utilizado para preeencher campos nulos com o primeiro valor não nulo de uma sequência de valores
* **TRATAMENTO DE TEXTO**

TIPOS :

* LOWER()
* UPPER()
* TRIM()
* REPLACE()

Exemplo 1: Corrija o primeiro elemento das queries abaixo utilizando os comandos de tratamento de texto para que o resultado seja sempre TRUE

select upper(‘São Paulo’) = ‘SÃO PAULO’

select lower(‘São Paulo’) = ‘são paulo’

select trim(‘SÃO PAULO ‘) = ‘SÃO PAULO’

select replace(‘SAO PAULO’, ‘SAO’, ‘SÃO’) = ‘SÃO PAULO)

sintaxe do replace = 1º a palavra, 2º a palavra que deseja substituir, 3º a nova que deseja

**RESUMO**

* UPPER() é utilizado para transformar todo texto em letras maiúsculas
* LOWER() é utilizado para transformar todo texto em letras minúsculas
* TRIM() é utilizado para remover todo os espaços das extremidades de um texto
* REPLACE() é utilizado para substituir uma string por outra string
* **TRATAMENTO DE DATAS**
* INTERVAL
* DATE\_TRUNC
* EXTRACT
* DATEDIFF

Exemplo 1: Soma de datas utilizando INTERVAL

– Calcule a data de hoje mais 10 unidades (dias, semanas, meses, horas)

select current\_date + 10

select current\_date + interval ‘10 weeks’

select (current\_date + interval ‘10 weeks’)::date

select (current\_date + interval ‘10 months’)::date

select (current\_date + interval ‘10 hours’)

Exemplo 2: Truncagem de datas utilizando DATE\_TRUNC

– Calcule quantas visitas ocorreram por mês no site da empresa

select visit\_page\_date, count(\*)

from sales.funnel

group by visit\_page\_date

order by visit\_page\_date desc

select

date\_trunc(‘month’, visit\_page\_date)::date as visit\_page\_month,

count(\*)

from sales.funnel

group by visit\_page\_month

order by visit\_page\_month desc

Exemplo 3: Extração de unidades de uma data utilizando EXTRACT

– Calcule qual é o dia da semana que mais recebe visitas ao site

select

current\_date,

extract(‘dow’ from current\_date)

dow = day of week, domingo = 0

select

extract(‘dow’ from visit\_page\_date) as dia\_da\_semana

count(\*)

from sales.funnel

group by dia\_da\_semana

order by dia\_da\_semana

Exemplo 4: Diferença entre datas com o operador de subtração (-)

– Calcule a diferença entre hoje e ‘2018-06-01’, em dias, semanas, meses e anos.

select (current\_date - ‘2018-06-01’)

select (current\_date - ‘2018-06-01’)/7

select (current\_date - ‘2018-06-01’)/30

select (current\_date - ‘2018-06-01’)/365

ou

select datediff(‘weeks’, ‘2018-06-01’, current\_date)

**RESUMO**

* O comando INTERVAL é utilizado para somar datas na unidade desejada. Caso a unidade não seja informada, o SQL irá entender que a soma foi feita em dias.
* O comando DATE\_TRUNC é utilizado para truncar uma data no início do período
* O comando EXTRACT é utilizado para extrair unidades de uma data / timestamp
* O cálculo da diferença entre datas com o operador de subtração (-) retorna valores em dias. Para calcular a diferença entra datas em outra unidade é necessário fazer uma transformação de unidades (ou criar uma função para isso)
* **FUNÇÕES**
* Servem para criar comandos personalizados de scripts usados recorrentemente.

Exemplo 1: Crie uma função chamada DATEDIFF para calcular a diferença entre duas datas em dias, semanas, meses, anos

select datediff(‘weeks’, ‘2018-06-01’, current\_date)

create function datediff(unidade varchar, data\_inicial date, data\_final date)

returns integer

language sql

as

$$

select

case

when unidade in (‘d’, ‘day’, ‘days’) then (data\_final - data\_inicial)

when unidade in (‘w’, ‘week’, ‘weeks’) then (data\_final - data\_inicial) / 7

when unidade in (‘m’, ‘month’, ‘months’) then (data\_final - data\_inicial) / 30

when unidade in (‘y’, ‘year’, ‘years’) then (data\_final - data\_inicial) / 365

end as diferenca

$$

select datediff(‘day’, ‘2021-02-04’, current\_date)

Exemplo 2: Delete a função DATEDIFF criada no exercício anterior

drop function datediff

**RESUMO**

* Para criar funções, utiliza-se o comando CREATE FUNCTION
* Para que o comando funcione é obrigatório informar (a) quais as unidades dos INPUTS (b) quais as unidades dos OUTPUTS e © em qual linguagem a função será escrita
* O script da função deve estar delimitado por $$ $$
* Para deletar uma função utiliza-se o comando DROP FUNCTION