

# SQL - FUNÇÕES DE UMA ÚNICA LINHA

---

**SIMONE DOMINICO - AULA 3**

---

# FUNÇÕES SQL

- ▶ São um recurso avançado do SQL e podem ser utilizadas para:
  - ▶ Executar cálculos usando dados.
  - ▶ Modificar itens de dados individuais.
  - ▶ Manipular saída para grupos de linhas.
  - ▶ Formatar datas e números para exibição.
  - ▶ Converter tipos de dados de coluna.

---

# TIPOS DE FUNÇÕES

- ▶ Funções de Datas;
- ▶ Funções numéricas;
- ▶ Funções de Caracter;
- ▶ Funções de Conversão;

---

# FUNÇÕES DE DATAS – AGE()

- ▶ Calcula idades.
- ▶ **age(timestamp, timestamp)**

```
tpch=> SELECT age('2017-01-01', '2011-06-24');
```

---

# FUNÇÕES DE DATAS – AGE()

- ▶ Calcula idades.
- ▶ **age(timestamp, timestamp)**

```
tpch=> SELECT age('2017-01-01', '2011-06-24');
```

```
      age  
-----  
5 years 6 mons 7 days  
(1 row)
```

---

# FUNÇÕES DE DATAS – AGE()

```
tpch=> select l_shipdate, l_commitdate,  
age(l_shipdate, l_commitdate) from linei  
tem where l_shipdate='1993-01-29';
```

# FUNÇÕES DE DATAS – AGE()

```
tpch=> select l_shipdate, l_commitdate,  
age(l_shipdate, l_commitdate) from linei  
tem where l_shipdate='1993-01-29';
```

l_shipdate	l_commitdate	age
1993-01-29	1993-02-28	-30 days
1993-01-29	1992-11-27	2 mons 2 days
1993-01-29	1993-01-09	20 days
1993-01-29	1992-11-21	2 mons 8 days
1993-01-29	1993-01-19	10 days
1993-01-29	1992-12-05	1 mon 24 days
1993-01-29	1993-03-17	-1 mons -19 days
1993-01-29	1992-11-24	2 mons 5 days

---

# FUNÇÕES DE DATAS – AGE()

- ▶ Calcula idades.
- ▶ **age(timestamp)**

```
tpch=> SELECT current_date,  
            AGE(timestamp '2000-01-01');
```



---

# FUNÇÕES DE DATAS – AGE()

- ▶ Calcula idades.
- ▶ **age(timestamp)**

```
tpch=> SELECT current_date,  
            AGE(timestamp '2000-01-01');
```

```
date | age  
-----+-----  
2018-08-24 | 18 years 7 mons 23 days  
(1 row)
```

---

# FUNÇÕES DE DATAS – CURRENT\_DATE

## ▶ DATA ATUAL

```
tpch=> SELECT current_date;
```

---

# FUNÇÕES DE DATAS – CURRENT\_DATE

## ▶ DATA ATUAL

```
tpch=> SELECT current_date;
```

```
      date  
-----  
2018-08-24  
(1 row)
```

---

# FUNÇÕES DE DATAS – CURRENT\_TIME

## ▶ HORA ATUAL

```
tpch=> SELECT current_time;  
          timetz  
-----  
00:29:55.489265-03  
(1 row)
```

---

# FUNÇÕES DE DATAS – CURRENT\_TIMESTAMP()

- ▶ HORA e DATA ATUAL - precisão

```
tpch=> SELECT current_timestamp;  
          now  
-----  
2018-08-24 00:32:45.041207-03  
(1 row)
```

---

# FUNÇÕES DE DATAS-EXTRACT

- ▶ Extrai informações de uma data

```
tpch=> SELECT extract(year from current_date);  
date_part  
-----  
          2018  
(1 row)
```

---

# FUNÇÕES DE DATAS-DATE\_PART

- ▶ Extrai informações de um campo data

```
tpch=> SELECT date_part('hour',TIMESTAMP '2017-03-18 10:20:30') h,  
           date_part('minute',TIMESTAMP '2017-03-18 10:20:30') m,  
           date_part('second',TIMESTAMP '2017-03-18 10:20:30') s;
```

# FUNÇÕES DE DATAS-DATE\_PART

```
tpch=> SELECT date_part('hour',TIMESTAMP '2017-03-18 10:20:30') h,  
           date_part('minute',TIMESTAMP '2017-03-18 10:20:30') m,  
           date_part('second',TIMESTAMP '2017-03-18 10:20:30') s;
```

```
h | m | s  
--+-+--  
10 | 20 | 30  
(1 row)
```



---

# FUNÇÕES DE DATAS - DATE\_TRUNC

```
tpch=> select l_shipdate, l_commitdate, date_trunc('year', l_
commitdate) from lineitem where l_shipdate='1993-01-29';
```

# FUNÇÕES DE DATAS - DATE\_TRUNC

```
tpch=> select l_shipdate, l_commitdate, date_trunc('year', l_
commitdate) from lineitem where l_shipdate='1993-01-29';
```

l_shipdate	l_commitdate	date_trunc
1993-01-29	1993-02-28	1993-01-01 00:00:00-02
1993-01-29	1992-11-27	1992-01-01 00:00:00-02
1993-01-29	1993-01-09	1993-01-01 00:00:00-02

---

# FUNÇÕES DE DATAS-TO\_DATE

- ▶ Converte string para o formato data.

```
tpch=> SELECT to_date('20180825', 'YYYYMMDD');  
to_date  
-----  
2018-08-25
```

# FUNÇÕES NUMÉRICAS - SQL

---

---

# FUNÇÕES NUMÉRICAS – ABS

- ▶ **Calcula o valor absoluto de um número.**

```
tpch=> select abs(-19.2);  
abs  
-----  
19.2  
(1 row)
```

---

# FUNÇÕES NUMÉRICAS – CBRT

- ▶ **Calcula a raiz cúbica**

```
tpch=> select p_retailprice,  
cbirt(p_retailprice) from part  
;
```

# FUNÇÕES NUMÉRICAS – CBRT

- ▶ **Calcula a raiz cúbica**

```
tpch=> select p_retailprice,  
cbirt(p_retailprice) from part  
;
```

p_retailprice	cbirt
901.00	9.65846840908043
902.00	9.66204032819639
903.00	9.66560960828766

---

# FUNÇÕES NUMÉRICAS – CEIL

- ▶ **O menor inteiro não menor que o argumento**

```
tpch=> select p_retailprice,  
ceil(p_retailprice) from part  
;
```



# FUNÇÕES NUMÉRICAS – CEIL

- ▶ **O menor inteiro não menor que o argumento**

```
tpch=> select p_retailprice,  
ceil(p_retailprice) from part  
;
```

p_retailprice	ceil
901.00	901
902.00	902
903.00	903

---

# FUNÇÕES NUMÉRICAS – MOD

- ▶ **resto de  $y/x$**

```
tpch=> select p_retailprice,  
mod(p_retailprice,p_size) fro  
m part;
```

---

# FUNÇÕES NUMÉRICAS – MOD

## ▶ resto de $y/x$

```
tpch=> select p_retailprice,  
mod(p_retailprice,p_size) fro  
m part;
```

p_retailprice	mod
901.00	5.00
902.00	0.00
903.00	0.00

---

# FUNÇÕES NUMÉRICAS – FLOOR

- ▶ **O maior inteiro não maior que o argumento**

```
tpch=> select p_retailprice,  
floor(p_retailprice) from part;
```

# FUNÇÕES NUMÉRICAS – FLOOR

- ▶ **O maior inteiro não maior que o argumento**

```
tpch=> select p_retailprice,  
floor(p_retailprice) from part;
```

p_retailprice	floor
901.00	901
902.00	902
903.00	903
904.00	904

---

# FUNÇÕES NUMÉRICAS – ROUND

- ▶ **Arredonda um valor numérico para o número inteiro mais próximo ou um número com o número de casas decimais.**

---

# FUNÇÕES NUMÉRICAS – ROUND

```
tpch=> SELECT  
        ROUND( 10.817, 2 );
```

```
round  
-----  
10.82  
(1 row)
```

---

# FUNÇÕES NUMÉRICAS – TRUNC

- ▶ **Retorna um número truncado para um número inteiro ou truncado para as casas decimais especificadas.**

```
tpch=> select p_retailprice,  
trunc(p_retailprice, -2) from  
part;
```



# FUNÇÕES NUMÉRICAS - TRUNC

```
tpch=> select p_retailprice,  
trunc(p_retailprice, -2) from  
part;
```

p_retailprice	trunc
901.00	900
902.00	900
903.00	900

---

# FUNÇÕES NUMÉRICAS – SQRT

- ▶ **Raiz quadrada.**

```
tpch=> select p_retailprice,  
sqrt(p_retailprice) from part  
;
```

# FUNÇÕES NUMÉRICAS – SQRT

## ► Raiz quadrada.

```
tpch=> select p_retailprice,  
sqrt(p_retailprice) from part  
;
```

p_retailprice	sqrt
901.00	30.016662039607269
902.00	30.0333314835362413
903.00	30.049958402633439
904.00	30.066592756745817

# FUNÇÕES DE CARACTERES - SQL

---

---

# FUNÇÕES DE CHARACTER – CONCAT

- ▶ **Concatena duas ou mais strings.**

```
tpch=> select concat(p_name,p_type) as  
NewName from part;
```

---

# FUNÇÕES DE CHARACTER – CONCAT

- ▶ **Concatena duas ou mais strings.**

```
tpch=> select concat(p_name,p_type) as  
NewName from part;
```

newname

-----  
goldenrod lavender spring chocolate lacePROMO BURNISHED COPPER  
blush thistle blue yellow saddleLARGE BRUSHED BRASS  
spring green yellow purple cornsilkSTANDARD POLISHED BRASS  
cornflower chocolate smoke green pinkSMALL PLATED BRASS  
forest brown coral puff creamSTANDARD POLISHED TIN

---

# FUNÇÕES DE CHARACTER – CONCAT

- ▶ **Concatena duas ou mais strings.**

```
tpch=> select concat(p_name, ' ', p_type) as  
NewName from part;
```

---

# FUNÇÕES DE CHARACTER – CONCAT

- ▶ **Concatena duas ou mais strings.**

```
tpch=> select concat(p_name, ' ', p_type) as  
NewName from part;
```

newname

---

```
goldenrod lavender spring chocolate lace PROMO BURNISHED COPPER  
blush thistle blue yellow saddle LARGE BRUSHED BRASS  
spring green yellow purple cornsilk STANDARD POLISHED BRASS  
cornflower chocolate smoke green pink SMALL PLATED BRASS  
forest brown coral puff cream STANDARD POLISHED TIN
```



---

# FUNÇÕES DE CARACTER – FORMAT

- ▶ **Formata argumentos baseados em uma string de formato.**

```
tpch=> select format('NOME: %s, TIPO: %s', p_name, p_type) as NewName from part;
```

---

# FUNÇÕES DE CHARACTER – FORMAT

```
tpch=> select format('%s nome, %s tipo'
,p_name,p_type) as NewName from part;
```

newname

---

NOME: goldenrod lavender spring chocolate lace, TIPO: PROMO BURNISHED COPPER

NOME: blush thistle blue yellow saddle, TIPO: LARGE BRUSHED BRASS

NOME: spring green yellow purple cornsilk, TIPO: STANDARD POLISHED BRASS

---

# FUNÇÕES DE CHARACTER - LEFT

- ▶ **Retorna os N primeiros caracteres.**

```
tpch=> select left(r_name,3) from region;
```

---

# FUNÇÕES DE CHARACTER - LEFT

- ▶ **Retorna os N primeiros caracteres.**

```
tpch=> select left(r_name,3) from region;
```

```
left
-----
AFR
AME
ASI
EUR
MID
(5 rows)
```

---

# FUNÇÕES DE CHARACTER – LENGTH

- ▶ **Retorna o número de caracteres ou bytes de uma string.**

```
tpch=> select r_name, length(r_name)
from region;
```

---

# FUNÇÕES DE CHARACTER – LENGTH

- ▶ **Retorna o número de caracteres ou bytes de uma string.**

```
tpch=> select r_name, length(r_name)
from region;
```

r_name	length
AFRICA	6
AMERICA	7
ASIA	4
EUROPE	6
MIDDLE EAST	11

(5 rows)

---

# FUNÇÕES DE CHARACTER – POSITION

- ▶ **Retorna a posição de uma substring em uma string.**

```
tpch=> select r_name, position('A' IN  
r_name) from region;
```

---

# FUNÇÕES DE CHARACTER – POSITION

```
tpch=> select r_name, position('A' IN  
r_name) from region;
```

r_name	position
AFRICA	1
AMERICA	1
ASIA	1
EUROPE	0
MIDDLE EAST	9

(5 rows)



---

# FUNÇÕES DE CHARACTER – RIGHT

- ▶ **Retorna os n caracteres finais.**

```
tpch=> select r_name, right(r_name,  
3) from region;
```

---

# FUNÇÕES DE CHARACTER – RIGHT

- ▶ **Retorna os n caracteres finais.**

```
tpch=> select r_name, right(r_name,  
3) from region;
```

r_name	right
AFRICA	ICA
AMERICA	ICA
ASIA	SIA
EUROPE	OPE
MIDDLE EAST	AST

(5 rows)

---

# FUNÇÕES DE CHARACTER – REPLACE

- ▶ **Substitui todas as ocorrências de string.**

```
tpch=> select r_name, replace(r_name, 'A', '0') from region;
```

---

# FUNÇÕES DE CHARACTER – REPLACE

```
tpch=> select r_name, replace(r_name, 'A', '0') from region;
```

r_name	replace
AFRICA	0FRICO
AMERICA	0MERIC0
ASIA	0SIO
EUROPE	EUROPE
MIDDLE EAST	MIDDLE E0ST

(5 rows)

---

# FUNÇÕES DE CHARACTER – SUBSTRING

- ▶ **Extrai uma substring de uma string.**

```
tpch=> select r_name, substring(r_name,  
1, 3) from region;
```

---

# FUNÇÕES DE CHARACTER – SUBSTRING

- ▶ **Extrai uma substring de uma string.**

```
tpch=> select r_name, substring(r_name,  
1, 3) from region;
```

r_name	substring
AFRICA	AFR
AMERICA	AME
ASIA	ASI
EUROPE	EUR
MIDDLE EAST	MID

(5 rows)

# FUNÇÕES DE CONVERSÃO - SQL

---

---

# FUNÇÕES DE CONVERSÃO

**CONVERSÃO DE  
DADOS**

```
graph TD; A[CONVERSÃO DE DADOS] --- B[CONVERSÃO IMPLÍCITA DE DADOS]
```

**CONVERSÃO  
IMPLÍCITA DE  
DADOS**



---

# FUNÇÕES DE CONVERSÃO

**CONVERSÃO DE  
DADOS**

```
graph TD; A[CONVERSÃO DE DADOS] --- B[CONVERSÃO IMPLÍCITA DE DADOS]; A --- C[CONVERSÃO EXPLÍCITA DE DADOS];
```

**CONVERSÃO  
IMPLÍCITA DE  
DADOS**

**CONVERSÃO  
EXPLÍCITA DE  
DADOS**

---

# FUNÇÕES DE CONVERSÃO

**CONVERSÃO DE  
DADOS**

```
graph TD; A[CONVERSÃO DE DADOS] --- B[CONVERSÃO IMPLÍCITA DE DADOS]; A --- C[CONVERSÃO EXPLÍCITA DE DADOS];
```

**CONVERSÃO  
IMPLÍCITA DE  
DADOS**

**CONVERSÃO  
EXPLÍCITA DE  
DADOS**

---

# FUNÇÕES DE CONVERSÃO -

**TO\_NUMBER**

**TO\_DATE**

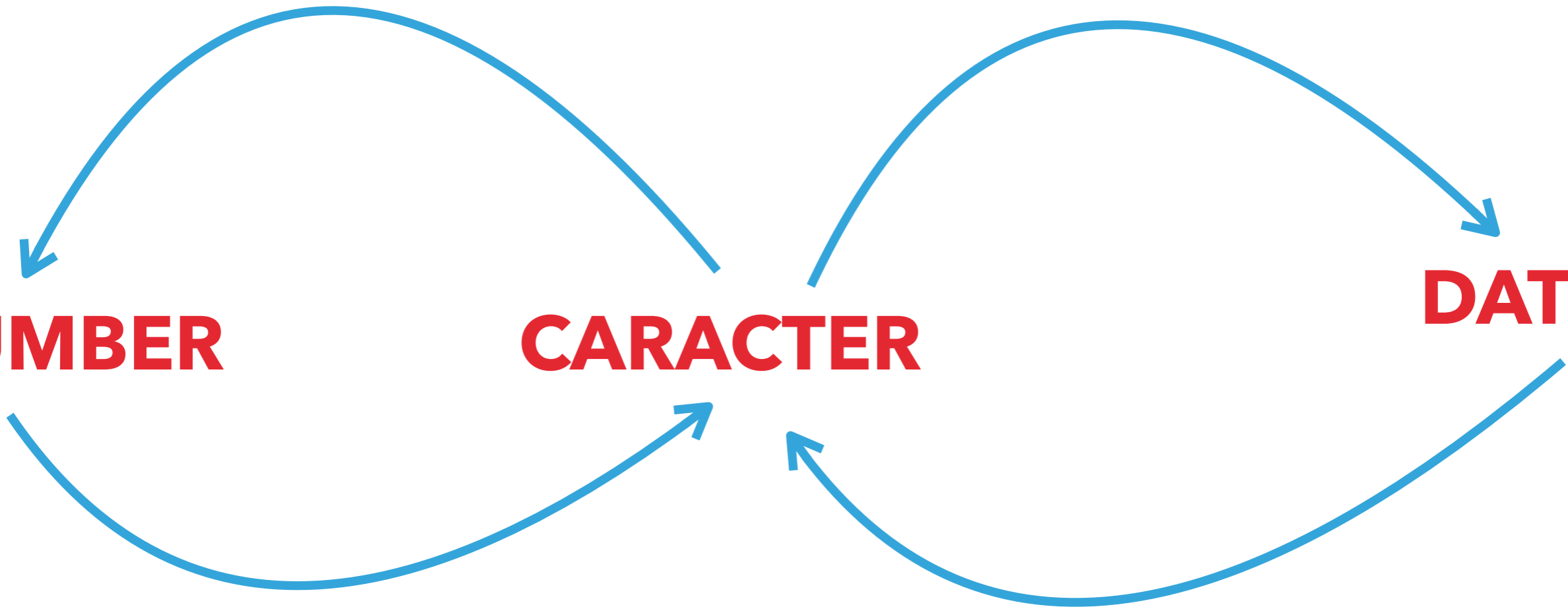
**NUMBER**

**CARACTER**

**DATE**

**TO\_CHAR**

**TO\_CHAR**



---

# FUNÇÕES DE CONVERSÃO – TO\_CHAR

```
tpch=> select l_extendedprice, to_char  
(l_extendedprice, '$9999.99') price fr  
om lineitem where l_extendedprice=9162  
.00; █
```



---

# FUNÇÕES DE CONVERSÃO – TO\_CHAR

```
tpch=> select l_shipdate, to_char(l_shipdate, 'DD-MON-YYYY') from lineitem where l_shipdate > '1998-11-18';
```

# FUNÇÕES DE CONVERSÃO - TO\_CHAR

```
tpch=> select l_shipdate, to_char(l_shipdate, 'DD-MON-YYYY') from lineitem where l_shipdate > '1998-11-18';
```

l_shipdate	to_char
1998-11-20	20-NOV-1998
1998-11-23	23-NOV-1998
1998-11-23	23-NOV-1998
1998-11-26	26-NOV-1998
1998-11-21	21-NOV-1998

---

# FUNÇÕES DE CONVERSÃO – **TO\_CHAR**

[http://pgdocptbr.sourceforge.net/pg80/  
functions-formatting.html](http://pgdocptbr.sourceforge.net/pg80/functions-formatting.html)



---

# FUNÇÕES DE CONVERSÃO – TO\_NUMBER

```
tpch=> select to_number('$ 9162.00', '999999')  
;  
;
```

---

# FUNÇÕES DE CONVERSÃO – TO\_NUMBER

```
tpch=> select to_number('$ 9162.00', '999999')  
;  
;
```

```
to_number  
-----  
          9162  
(1 row)
```

---

# FUNÇÕES DE CONVERSÃO – TO\_DATE

```
tpch=> SELECT to_date('20170103', 'YYYY  
MMDD');
```

---

# FUNÇÕES DE CONVERSÃO – TO\_DATE

```
tpch=> SELECT to_date('20170103','YYYYMMDD');
```

```
to_date
-----
2017-01-03
(1 row)
```

---

# FUNÇÕES DE CONVERSÃO – TO\_DATE

```
tpch=> SELECT to_date('2017 Feb 20', 'Y  
YYY Mon DD');
```

---

# FUNÇÕES DE CONVERSÃO – TO\_DATE

```
tpch=> SELECT to_date('2017 Feb 20', 'Y  
YYY Mon DD');
```

```
to_date  
-----  
2017-02-20  
(1 row)
```

---

# ANINHANDO FUNÇÕES

```
tpch=> select c_name, to_char(cbrt(c_acct  
bal), '$ 999.99') VALOR from customer;
```

# ANINHANDO FUNÇÕES

```
tpch=> select c_name, to_char(cbrt(c_acctbal), '$ 999.99') VALOR from customer;
```

c_name	valor
Customer#0000000001	\$ 8.93
Customer#0000000002	\$ 4.95
Customer#0000000003	\$ 19.57
Customer#0000000004	\$ 14.21
Customer#0000000005	\$ 9.26
Customer#0000000006	\$ 19.69
Customer#0000000007	\$ 21.23



# EXERCÍCIOS

---

# EXERCÍCIOS

1. Crie uma consulta para exibir a data corrente.

# EXERCÍCIOS

2. Crie uma consulta para exibir o nome (c\_name) e a raiz quadrada da conta do cliente (customer) do país (c\_nationkey) 24 e com conta (c\_acctbal) maior que R\$ 9990

c_name	sqrt
Customer#000002487	99.998599990199863
Customer#000010494	99.994349840378481
Customer#000015607	99.989199416737006
Customer#000027412	99.994249834677994
Customer#000089748	99.961942758231745

11 ROWS

# EXERCÍCIOS

2. Altere a consulta do exercício 2 para exibir a raiz quadrada em formato \$99.00 e a coluna nomeada como VALOR.

c_name	valor
Customer#000002487	\$ 100.00
Customer#000010494	\$ 99.99
Customer#000015607	\$ 99.99
Customer#000027412	\$ 99.99
Customer#000089748	\$ 99.96

# EXERCÍCIOS

4. Altere a consulta do exercício 3 para exibir a quantidade de caracteres na coluna `c_comment`.

<code>c_name</code>	<code>valor</code>	<code>length</code>
Customer#000002487	\$ 100.00	73
Customer#000010494	\$ 99.99	103
Customer#000015607	\$ 99.99	52
Customer#000027412	\$ 99.99	58
Customer#000089748	\$ 99.96	99

# EXERCÍCIOS

5. Crie uma consulta para exibir o nome (c\_name) e o telefone do cliente(customer), e o telefone no formato numérico do país (c\_nationkey) 24 e com conta (c\_acctbal) maior que R\$ 9990

c_name	c_phone	to_number
Customer#000002487	34-430-662-5641	34430662564
Customer#000010494	34-533-497-3882	34533497388
Customer#000015607	34-714-539-7404	34714539740
Customer#000027412	34-426-311-1651	34426311165
Customer#000089748	34-414-771-9804	34414771980
Customer#000104503	34-973-102-1923	34973102192