

```
import polars as pl

# Criando DataFrames
clientes = pl.DataFrame({
    "cliente_id": [1, 2, 3, 4],
    "nome": ["Ana", "Bruno", "Clara", "Daniel"]
})

print(clientes)
```

shape: (4, 2)

cliente_id	nome
---	---
i64	str
1	Ana
2	Bruno
3	Clara
4	Daniel

```
pedidos = pl.DataFrame({
    "pedido_id": [101, 102, 103, 104, 105],
    "cliente_id": [1, 2, 3, 1, 5],
    "valor": [100.50, 250.75, 75.00, 130.00, 79.00]
})

print(pedidos)
```

shape: (5, 3)

pedido_id	cliente_id	valor
---	---	---
i64	i64	f64
101	1	100.5
102	2	250.75
103	3	75.0
104	1	130.0
105	5	79.0

```
res_ij = clientes.join(pedidos, on="cliente_id", how="inner")
print(res_ij)
```

shape: (4, 4)

cliente_id	nome	pedido_id	valor
---	---	---	---
i64	str	i64	f64
1	Ana	101	100.5
2	Bruno	102	250.75
3	Clara	103	75.0
1	Ana	104	130.0

```
res_lj = clientes.join(pedidos, on="cliente_id", how="left")
print(res_lj)
```

shape: (5, 4)

cliente_id	nome	pedido_id	valor
---	---	---	---
i64	str	i64	f64
1	Ana	101	100.5
1	Ana	104	130.0
2	Bruno	102	250.75
3	Clara	103	75.0
4	Daniel	null	null

```
res_rj = clientes.join(pedidos, on="cliente_id", how="right")
print(res_rj)
```

shape: (5, 4)

nome	pedido_id	cliente_id	valor
---	---	---	---
str	i64	i64	f64



Ana	101	1	100.5
Bruno	102	2	250.75
Clara	103	3	75.0
Ana	104	1	130.0
null	105	5	79.0

```
res_oj = clientes.join(pedidos, on="cliente_id", how="full")
print(res_oj)
```

shape: (6, 5)

cliente_id	nome	pedido_id	cliente_id_right	valor
---	---	---	---	---
i64	str	i64	i64	f64
1	Ana	101	1	100.5
2	Bruno	102	2	250.75
3	Clara	103	3	75.0
1	Ana	104	1	130.0
null	null	105	5	79.0
4	Daniel	null	null	null

/tmp/ipython-input-3166284463.py:1: DeprecationWarning: Use of `how='outer'` should be replaced with `how='full'`.  
 res\_oj = clientes.join(pedidos, on="cliente\_id", how="outer")

```
res_cj = clientes.join(pedidos, how="cross")
print(res_cj)
```

shape: (20, 5)

cliente_id	nome	pedido_id	cliente_id_right	valor
---	---	---	---	---
i64	str	i64	i64	f64
1	Ana	101	1	100.5
1	Ana	102	2	250.75
1	Ana	103	3	75.0
1	Ana	104	1	130.0
1	Ana	105	5	79.0
...	...	...	...	...
4	Daniel	101	1	100.5
4	Daniel	102	2	250.75
4	Daniel	103	3	75.0
4	Daniel	104	1	130.0
4	Daniel	105	5	79.0

```
res = res_ij.group_by(["nome", "cliente_id"]).agg(pl.col("valor").mean())
print(res)
```

shape: (3, 3)

nome	cliente_id	valor
---	---	---
str	i64	f64
Ana	1	115.25
Bruno	2	250.75
Clara	3	75.0

```
res = (res_oj.with_columns(pl.col("valor") > 100)
      .group_by("nome")
      .agg(pl.col("valor").sum()))
print(res)
```

shape: (5, 2)

nome	valor
---	---
str	u32
Clara	0
Ana	2
Daniel	0
null	0
Bruno	1

```
vendas = pl.DataFrame({
    "id_venda": [1, 2, 3],
    "id_cl": [1, 2, 1],
    "id_prod": [101, 102, 103],
```

```
        "qtde": [2, 1, 1]
    })

    detalhes_pedidos = pl.DataFrame({
        "id_ped": [201, 202, 203],
        "cl_id": [1, 2, 1],
        "id_prod": [101, 102, 104],
        "valor": [50.00, 75.00, 100.00]
    })
```

```
final = vendas.join(detalhes_pedidos,
                    left_on = ["id_cl", "id_prod"],
                    right_on = ["cl_id", "id_prod"],
                    how = "inner")

print(final)
```

shape: (2, 6)

id_venda	id_cl	id_prod	qtde	id_ped	valor
---	---	---	---	---	---
i64	i64	i64	i64	i64	f64
1	1	101	2	201	50.0
2	2	102	1	202	75.0

```
!pip install pytz
```

Requirement already satisfied: pytz in /usr/local/lib/python3.12/dist-packages (2025.2)

```
from datetime import datetime
import pytz

# Define o fuso horário de Brasília
brasilia_timezone = pytz.timezone('America/Sao_Paulo')

# Obtém a data e hora atuais no fuso horário de Brasília
now_brasilia = datetime.now(brasilia_timezone)

# Imprime a data e hora com o fuso horário
print("Data e hora atuais (Brasília):", now_brasilia)

# Opcional: Imprime em um formato específico
print("Data e hora formatadas (Brasília):", now_brasilia.strftime("%Y-%m-%d %H:%M:%S %Z%z"))
```

Data e hora atuais (Brasília): 2025-10-02 10:36:27.394128-03:00  
Data e hora formatadas (Brasília): 2025-10-02 10:36:27 -03-0300