# Thay đổi cấu trúc `DataFrame`

## 1 Giới thiệu

Thay đổi cấu trúc `DataFrame` để chỉ những thao tác làm thay đổi số dòng, cột của `DataFrame`.

[1]:
```
# DataFrame mẫu
import numpy as np
import pandas as pd

date = np.datetime64('2022-11-24', 'D') + np.random.randint(-100,␣
  ↪100, size = 100)
product = np.random.choice(['Apple', 'Banana', 'Cherry'], size = 100)
quantity = np.random.randint(100, size = 100)

df1 = pd.DataFrame({'Date' : date, 'Product' : product, 'Quantity' :␣
  ↪quantity})
df2 = pd.DataFrame(
    [['Apple', 10], ['Banana', 15], ['Cherry', 20]],
    columns = ['Product', 'Price']
)
```

[2]:
```
df1.head()
```

[2]:

| | Date | Product | Quantity |
|---|---|---|---|
| 0 | 2022-10-05 | Cherry | 8 |
| 1 | 2022-10-26 | Banana | 43 |
| 2 | 2022-10-06 | Cherry | 59 |
| 3 | 2022-11-01 | Apple | 49 |
| 4 | 2022-10-14 | Apple | 0 |

[3]:
```
df2
```

[3]:

| | Product | Price |
|---|---|---|
| 0 | Apple | 10 |
| 1 | Banana | 15 |
| 2 | Cherry | 20 |

## 2 Thêm cột mới

Để thêm một cột mới vào `DataFrame`, ta dùng phương thức `.assign()` như sau:

```
.assign(
    <tên_cột_mới> = <giá_trị>
)
```

Trong đó, `giá_trị` có thể là một `list`, `pandas.Series`.

[4]:
```
# Ví dụ:
s = pd.Series([1, 2, 3])
df2.assign(new = s)
```

[4]:
```
  Product  Price  new
0   Apple     10    1
1  Banana     15    2
2  Cherry     20    3
```

Có thể dùng `.assign()` để thêm nhiều cột mới cùng một lúc.

[5]:
```
# Ví dụ:
s2 = ['a', 'b', 'c']
df2.assign(
    new_1 = s2,
    new_2 = s2
)
```

[5]:
```
  Product  Price new_1 new_2
0   Apple     10     a     a
1  Banana     15     b     b
2  Cherry     20     c     c
```

Có thể xem tài liệu cùng với các ví dụ khác tại đây.

## 3 Thêm dòng mới vào `DataFrame`

Giả sử có 2 `DataFrame` là `df1` và `df2`, để thêm dòng từ vào `df2` vào `df1`, có thể dùng hàm `pd.concat()` với cú pháp:

```
pd.concat(
    [df1, df2],
    ignore_index
)
```

Tham số `ignore_index` dùng để báo cho `pandas` cách đánh `label` của dòng mới. Có hai giá trị là `True` và `False`. `True` sẽ bỏ qua `label` (nếu có) của các dòng mới, `False` sẽ sử dụng `label` (nếu có) của các dòng mới. Mặc định là `False`.

[6]:
```
# Ví dụ, ignore_index = False
pd.concat([df2, df2])
```

[6]:
```
  Product  Price
0   Apple     10
1  Banana     15
2  Cherry     20
0   Apple     10
1  Banana     15
2  Cherry     20
```

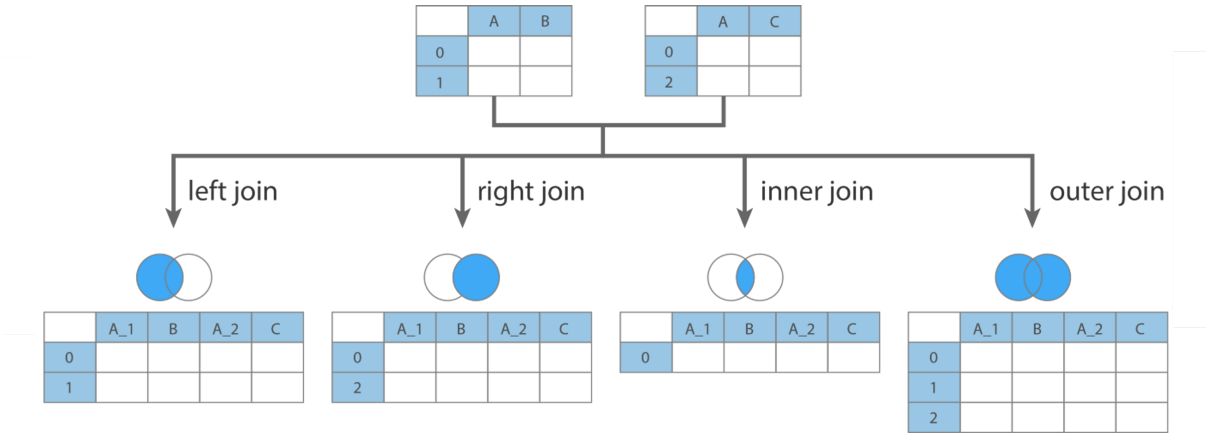
[7]:
```
# ignore_index = True
df3 = pd.DataFrame(
    ['Durian', 'Fig'],
    columns = ['Product']
)
pd.concat(
    [df2, df3],
    ignore_index = True
)
```

[7]:
```
  Product  Price
0   Apple   10.0
1  Banana   15.0
2  Cherry   20.0
3  Durian    NaN
4     Fig    NaN
```

Có thể xem tài liệu cùng các ví dụ khác tại đây.

# 4 Nối hai DataFrame

Nối hai `DataFrame` là hành động bổ sung thông tin vào `df_trái` từ `df_phải` dựa trên một hai nhiều cột chung nào đó giữa `df_trái` và `df_phải`.




Minh họa việc nối 2 DataFrame theo nhiều cách

Để thực hiện việc nối hai `DataFrame` có chung **một cột**, dùng hàm `pd.merge()` như sau:

```
pd.merge(df_trái, df_phải, how)
```

Tham số `how` dùng để chỉ ra cách thức `merge`, có 4 giá trị:

- `'inner'`: chỉ lấy các giá trị có trong cả hai bên trái, phải của `merge`, đây là giá trị mặc định.
- `'outer'`: lấy tất cả giá trị trong cả hai bên của `merge`.
- `'left'`: giữ nguyên tất cả các dòng bên `df_trái`.
- `'right'`: giữ nguyên tất cả các dòng bên `df_phải`.

Chi tiết đầy đủ có thể xem tại đây.

[8]:
```
# Ví dụ
print(df1.head(),'\n', df2)
print(pd.merge(df1, df2).head())
```

```
        Date Product  Quantity
0 2022-10-05  Cherry         8
1 2022-10-26  Banana        43
2 2022-10-06  Cherry        59
3 2022-11-01   Apple        49
4 2022-10-14   Apple         0
   Product  Price
0   Apple     10
1  Banana     15
2  Cherry     20
        Date Product  Quantity  Price
0 2022-10-05  Cherry         8     20
1 2022-10-06  Cherry        59     20
2 2022-09-27  Cherry        83     20
3 2022-08-27  Cherry        63     20
4 2022-09-26  Cherry        27     20
```

# 5 Xoá dòng, cột trong `DataFrame`

Để xoá dòng, cột trong `DataFrame`, dùng phương thức `.drop()` như sau:

```
.drop(
    index = <danh_sách_tên_dòng>,
    columns = <danh_sách_tên_cột>
)
```

Có thể bỏ qua `index` (hoặc `columns`) nếu chỉ muốn xóa cột (hoặc dòng).

[9]:
```
# ví dụ
df2.drop(
    index = [1],
```

```
    columns = ['Product']
)
```

[9]:
```
   Price
0     10
2     20
```

[10]:
```
# chỉ xóa cột
df2.drop(columns = ['Product'])
```

[10]:
```
   Price
0     10
1     15
2     20
```