Concatenating the dataset

In [2]:

```python
# Loading the library
import pandas as pd
```

In [4]:

```python
# Creating two dataframes
df1 = pd.DataFrame({
    'A': ['A0', 'A1', 'A2', 'A3'],
    'B': ['B0', 'B1', 'B2', 'B3'],
    'C': ['C0', 'C1', 'C2', 'C3'],
    'D': ['D0', 'D1', 'D2', 'D3']
})

df2 = pd.DataFrame({
    'A': ['A4', 'A5', 'A6', 'A7'],
    'B': ['B4', 'B5', 'B6', 'B7'],
    'C': ['C4', 'C5', 'C6', 'C7'],
    'D': ['D4', 'D5', 'D6', 'D7']
})
```

In [6]:

```python
df1
```

Out[6]:

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

In [8]:

```python
df2
```

Out[8]:

	A	B	C	D
0	A4	B4	C4	D4
1	A5	B5	C5	D5
2	A6	B6	C6	D6
3	A7	B7	C7	D7

In [10]:

```python
# Concatenating the dataframes
result = pd.concat([df1, df2])

result
```

Out[10]:

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
0	A4	B4	C4	D4
1	A5	B5	C5	D5

2	A6	B6	C6	D6
3	A7	B7	C7	D7

Merging the pandas dataframe on key

In [13]:

```
# Create two dataframes
df1 = pd.DataFrame({
    'key': ['A', 'B', 'C', 'D'],
    'value_df1': [1, 2, 3, 4]
})

df2 = pd.DataFrame({
    'key': ['C', 'D', 'E', 'F'],
    'value_df2': [5, 6, 7, 8]
})
```

In [15]:

```
df1
```

Out[15]:

	key	value_df1
0	A	1
1	B	2
2	C	3
3	D	4

In [17]:

```
df2
```

Out[17]:

	key	value_df2
0	C	5
1	D	6
2	E	7
3	F	8

In [19]:

```
# Merge the dataframes on the 'key' column
merged_df = pd.merge(df1, df2, on='key')

print(merged_df)
```

```
  key  value_df1  value_df2
0   C          3          5
1   D          4          6
```

Merging on multiple keys

In [22]:

```
# Create two dataframes
df1 = pd.DataFrame({
    'key1': ['A', 'B', 'C', 'D'],
    'key2': ['W', 'X', 'Y', 'Z'],
    'value_df1': [1, 2, 3, 4]
})

df2 = pd.DataFrame({
    'key1': ['B', 'C', 'C', 'E'],
    'key2': ['X', 'Y', 'Y', 'Z'],
    'value_df2': [5, 6, 7, 8]
})

# Merge the dataframes on 'key1' and 'key2'
merged_df = pd.merge(df1, df2, on=['key1', 'key2'])

print(merged_df)
```

```
  key1 key2  value_df1  value_df2
0    B    X          2          5
1    C    Y          3          6
2    C    Y          3          7
```

Join Operations

In [25]:

```
import pandas as pd

# Create two dataframes with different columns but a common index
df1 = pd.DataFrame({
    'A': ['A0', 'A1', 'A2'],
    'B': ['B0', 'B1', 'B2']
}, index=['K0', 'K1', 'K2'])

df2 = pd.DataFrame({
    'C': ['C0', 'C2', 'C3'],
    'D': ['D0', 'D2', 'D3']
}, index=['K0', 'K2', 'K3'])

# Join the dataframes using the default (left) join
result = df1.join(df2)

print(result)
```

```
     A   B    C    D
K0  A0  B0   C0   D0
K1  A1  B1  NaN  NaN
K2  A2  B2   C2   D2
```

In [27]:

```
#Note :
# left: use only keys from left frame.
# right: use only keys from right frame.
# outer: use union of keys from both frames.
# inner: use intersection of keys from both frames (default).
```

In [29]:

```
# If you want an outer join to include all indices, you can modify the join as:
result = df1.join(df2, how='outer')
result
```

Out[29]:

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

In [33]:

```
# If you want an right join to include only 2nd df , you can modify the join as:
result = df1.join(df2, how='right')
result
```

Out[33]:

	A	B	C	D
K0	A0	B0	C0	D0
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

In [ ]: