In [ ]:
```
import numpy as np
```

Bài số 16: Thư viện NumPy cơ bản

In [ ]:
```
# Bài 1: Viết chương trình sinh một ma trận 5×5 số nguyên ngẫu nhiên từ 0–9, sau
# tổng theo từng hàng.

A = np.random.randint(0, 10, size = (5,5))
print(A)
sum_rows = A.sum(axis = 1)
print("Tổng theo từng hàng: ", sum_rows)
```

```
[[3 0 8 8 5]
 [6 8 1 2 8]
 [3 8 9 4 6]
 [5 7 2 2 0]
 [6 0 6 4 3]]
Tổng theo từng hàng:  [24 25 30 16 19]
```

In [ ]:
```
# Bài 2: Tạo một ma trận ngẫu nhiên 4 × 4 và chuẩn hoá (giá trị trong khoảng [0,
A = np.random.rand(4, 4)
print(A)

print("\nChuẩn hóa A:")
A_norm = (A- A.min())/(A.max() - A.min())
print(A_norm)
```

```
[[0.7809761  0.3971864  0.95763582 0.19424517]
 [0.60891954 0.81951081 0.49692208 0.84830662]
 [0.97259309 0.69192337 0.76762758 0.78738848]
 [0.46082829 0.59791222 0.62869564 0.20232713]]

Chuẩn hóa A:
[[0.75381576 0.26073331 0.9807833  0.        ]
 [0.53276223 0.8033241  0.38887097 0.84032015]
 [1.         0.63940326 0.73666594 0.7620542 ]
 [0.34249866 0.51862032 0.55817002 0.01038348]]
```

In [ ]:
```
# Bài 3: Viết chương trình tạo một mảng số nguyên từ 1 đến 100, sau đó lấy ra tâ
# chẵn.
d = np.arange(1, 101, 1)
print(d)

d1 = d[d%2==0]
print("Các số chẵn trong mảng là: \n", d1)
```

```
[  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100]
Các số chẵn trong mảng là:
 [  2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34  36
  38  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68  70  72
  74  76  78  80  82  84  86  88  90  92  94  96  98 100]
```

In [ ]:
```
# Bài 4: Sinh một ma trận ngẫu nhiên 3 × 3, sau đó thay các giá trị lớn hơn 0.5
# còn lại thành 0.
A = np.random.rand(3, 3)
print(A)

A1 = np.where(A>0.5, 1, 0)
print("Ma trận sau khi đổi:\n ", A1)
```

```
[[0.95284681 0.49157014 0.32297673]
 [0.7425976  0.48654068 0.13611208]
 [0.59403149 0.61326502 0.56018676]]
Ma trận sau khi đổi:
  [[1 0 0]
 [1 0 0]
 [1 1 1]]
```

In [ ]:
```
# Bài 5: Viết hàm tính chuẩn Euclid (norm) của một vector bằng NumPy.
def euclidean_norm(v):
  return np.linalg.norm(v)

A = np.array([3,4])
print(euclidean_norm(A))
```

```
5.0
```

In [ ]:
```
# Bài 6: Sinh dữ liệu ngẫu nhiên 100 phần tử theo phân phối chuẩn, sau đó tính t
# phương sai và độ lệch chuẩn.
A = np.random.randn(100)
print(A)

print("Các phép tính với A:\n")
print("Trung bình: ", np.mean(A))
print("Trung vị: ", np.median(A))
print("Độ lệch chuẩn: ", np.std(A))
```

```
[-0.5203055   0.01649787 -0.37122481 -0.07944856 -0.00611419  2.28282546
 -2.06464413  0.20150586  1.45225017 -0.74647127  0.30501056  1.58216377
 -1.21129728 -0.1511808  -0.78582075 -0.70382535 -0.81904048 -0.61416618
  2.13385604  1.05312923 -0.42359708 -0.043556    0.38617428  0.76791287
 -0.29841805 -1.69457198  0.06353737 -0.25648811 -0.55549753 -0.52816127
  0.79171069  1.35599899 -0.58924216  0.69233901  1.3442837   0.64090655
  1.34495014 -1.01554168  0.71390364 -1.33999601 -1.48226134 -0.19154506
 -0.45926916  0.79630108 -0.34982234  2.14892461  1.04807435 -2.22709846
 -0.89621464 -0.51227394  1.17979499  0.34788339  0.51129651 -0.17705279
  0.02073904 -0.48455542 -1.11730329  0.17020228  0.26368543 -0.63774679
 -0.30843525  0.10934116 -1.8210726   0.716586   -1.2292674  -1.11684495
  0.44478355 -0.19672497  0.07036693  1.16542158  1.5227467   0.34565374
 -0.39421236  1.53389561  0.05652593  1.18947695 -0.25532796 -1.43858127
  0.93425299 -1.69847252 -0.05757452  0.27890283 -1.13784962 -1.30978054
  0.06437059  0.96048763  1.62121919 -1.78359807 -0.51418037  0.66069061
  0.34463742  1.02096496  1.60926042  0.62657391 -0.47274385  0.33967366
  1.17676268 -0.13272789  0.41754986  0.1414729 ]
Các phép tính với A:

Trung bình:  0.03746329143627922
Trung vị:  0.01861845492365369
Độ lệch chuẩn:  0.9770159669056293
```

In [ ]:
```
# Bài 7: Viết chương trình tạo một ma trận 10 × 10 với đường chéo chính bằng 1,
# chéo phụ bằng 2.
A = np.zeros((10, 10))
```

```
print(A)


print("Ma trận sau khi đổi:\n")
np.fill_diagonal(A, 1)

for i in range (10):
  A[i, 10-i-1] = 2

print(A)
```

```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
Ma trận sau khi đổi:

[[1. 0. 0. 0. 0. 0. 0. 0. 0. 2.]
 [0. 1. 0. 0. 0. 0. 0. 0. 2. 0.]
 [0. 0. 1. 0. 0. 0. 0. 2. 0. 0.]
 [0. 0. 0. 1. 0. 0. 2. 0. 0. 0.]
 [0. 0. 0. 0. 1. 2. 0. 0. 0. 0.]
 [0. 0. 0. 0. 2. 1. 0. 0. 0. 0.]
 [0. 0. 0. 2. 0. 0. 1. 0. 0. 0.]
 [0. 0. 2. 0. 0. 0. 0. 1. 0. 0.]
 [0. 2. 0. 0. 0. 0. 0. 0. 1. 0.]
 [2. 0. 0. 0. 0. 0. 0. 0. 0. 1.]]
```

In [ ]:
```
# Bài 8: Tạo mảng ngẫu nhiên 20 phần tử, sau đó sắp xếp tăng dần.
d = np.random.randn(20)
print(d)

d_sort = np.sort(d)
print(d_sort)
```

```
[-1.49729496  0.56297197 -0.58626395 -0.77302907 -0.51182903 -0.6881343
  0.90916275  1.1780089  -0.10213093 -2.40148477 -1.36931883 -0.09510911
  0.75315285 -0.45805155  0.23996167  1.60109867  1.16643204 -0.58394825
 -0.45962924  0.74485905]
[-2.40148477 -1.49729496 -1.36931883 -0.77302907 -0.6881343  -0.58626395
 -0.58394825 -0.51182903 -0.45962924 -0.45805155 -0.10213093 -0.09510911
  0.23996167  0.56297197  0.74485905  0.75315285  0.90916275  1.16643204
  1.1780089   1.60109867]
```

In [ ]:
```
# Bài 9: Viết hàm chuẩn hóa vector về độ dài 1 (unit vector).
def norm_vector(v):
  v_norm = (v-v.min())/(v.max() - v.min())
  return v_norm

v = np.array([1, 2, 3, 4, 5])
print(v)
print("v sau chuẩn hóa: ", norm_vector(v))
```

```
[1 2 3 4 5]
v sau chuẩn hóa:  [0.   0.25 0.5  0.75 1.  ]
```

In [ ]:
```
# Bài 10: Viết chương trình sinh ma trận 3 × 3, sau đó in ra nghịch đảo ma trận
A = np.random.rand(3, 3)
print(A)

A_inv = np.linalg.inv(A)
print(A_inv)
```

```
[[0.23871649 0.02112189 0.52888781]
 [0.90943493 0.91706882 0.91506036]
 [0.35926328 0.4261237  0.68304661]]
[[ 2.9200732   2.6048474  -5.75068185]
 [-3.61118201 -0.33285987  3.24208773]
 [ 0.71698643 -1.16241928  2.46612516]]
```

In [ ]:
```
# Bài 11: Viết chương trình tạo dữ liệu mô phỏng chiều cao (n=100) theo phân phố
# có trung bình 170 và độ lệch chuẩn 10.
A = np.random.normal(loc = 170, scale = 10, size = 100)
print("Dữ liệu mô phỏng chiều cao: ", A)
```

```
Dữ liệu mô phỏng chiều cao:  [157.35493786 170.30145991 176.21866073 164.48144604
166.26563314
 177.43179176 180.55898782 173.07637107 162.24817454 178.79959902
 172.92248031 169.38092314 170.57416916 156.43335231 165.74230285
 157.4221975  171.96265498 173.73360857 174.75731003 173.15038145
 167.13344843 169.9356724  158.72536126 160.87942552 170.68302474
 181.95953583 177.83139767 179.02160142 182.62961804 171.18112784
 158.18000672 171.886853   170.41913317 182.72318505 161.1394866
 178.28183019 178.40421886 195.25370432 160.67160599 152.60926429
 181.15673614 173.5721876  174.74419338 158.81440843 180.84579632
 165.19744019 179.74681167 171.16821669 188.49565787 178.04503983
 156.10136226 154.68600552 180.25797916 161.80005591 167.42704023
 165.19648584 157.68803231 177.45623292 172.1442322  176.7611843
 156.35354268 164.07106576 164.74174943 174.54336137 174.70437418
 156.94756892 162.81622164 177.14834519 163.87604309 175.05198849
 144.19444441 172.15825082 181.26040496 162.16411528 174.2553574
 176.04446906 169.90453848 168.3073     167.24755603 167.28325475
 170.8852649  168.2160677  167.55168818 200.10434359 171.48712271
 183.33402441 182.52932732 175.48203473 182.81208507 159.39933548
 179.25151942 176.79680695 194.56669013 162.37136279 189.85073641
 174.16290585 163.60374336 155.90103715 188.31166046 163.58752921]
```

In [ ]:
```
# Bài 12: Tạo mảng 2 chiều 5 × 5, sau đó thay đường viền ngoài bằng số 1, bên tr
# số 0.
n = 5
A = np.zeros((n,n), dtype = int)
print(A)

print("A sau khi thay:\n")
A[0,:] = 1
A[n-1,:] = 1
A[:,0] = 1
A[:,n-1] = 1
print(A)
```

```
[[0 0 0 0 0]
 [0 0 0 0 0]
 [0 0 0 0 0]
 [0 0 0 0 0]
 [0 0 0 0 0]]
A sau khi thay:

[[1 1 1 1 1]
 [1 0 0 0 1]
 [1 0 0 0 1]
 [1 0 0 0 1]
 [1 1 1 1 1]]
```

In [ ]:
```
# Bài 13: Viết chương trình tính tích Hadamard (phần tử nhân phần tử) giữa hai m
# 3 × 3.
A = np.random.rand(3,3)
print(A)
print("\n")
B = np.random.rand(3,3)
print(B)

print("A*B = ", A*B)
```

```
[[0.77468596 0.95220056 0.57717755]
 [0.1411716  0.95382102 0.06842384]
 [0.56342357 0.1215485  0.8175353 ]]


[[0.41990361 0.51926887 0.47040357]
 [0.63351453 0.89176043 0.38868167]
 [0.57013935 0.85219684 0.17555459]]
A*B =  [[0.32529343 0.49444811 0.27150638]
 [0.08943426 0.85057985 0.02659509]
 [0.32122995 0.10358325 0.14352208]]
[[1.25759796 1.74327397 0.83584412]
 [0.70254907 0.98219644 0.44915249]
 [0.77969538 1.09766147 0.45580221]]
```

In [ ]:
```
# Bài 14: Tạo mảng 2 chiều 6 × 6 ngẫu nhiên, sau đó tính tổng từng cột.
A = np.random.randint(0, 10, size=(6, 6))
print(A)

sum_cols = np.sum(A, axis = 0)
print(sum_cols)
```

```
[[7 4 3 0 2 0]
 [6 2 2 5 7 4]
 [7 1 8 3 9 5]
 [7 4 2 3 3 6]
 [6 0 8 7 2 3]
 [4 9 9 8 8 6]]
[37 20 32 26 31 24]
```

In [ ]:
```
# Bài 15: Viết chương trình xoay ma trận 3 × 3 90 độ.
# np.rot90(m, k=1, axes=(0, 1))
A = np.random.randint(0, 10, (3,3))
print(A)

A1= np.rot90(A, k = -1, axes=(0,1))
print(A1)
```

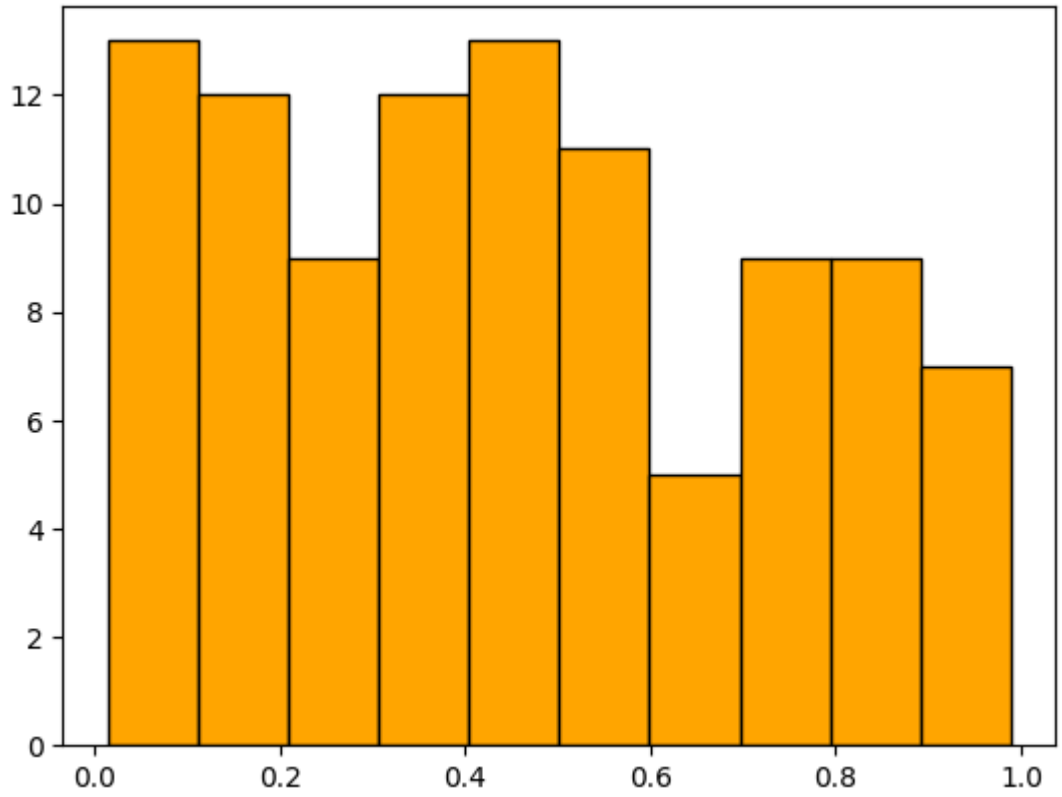
```
[[5 2 0]
 [1 1 2]
 [8 1 5]]
[[8 1 5]
 [1 1 2]
 [5 2 0]]
```

In [ ]:
```
import matplotlib.pyplot as plt
# plt.hist(data, bins=10, edgecolor='black', alpha=0.7)
```

In [ ]:
```
# Bài 16: Sinh mảng 100 phần tử từ phân phối đều [0,1], sau đó vẽ histogram bằng
A = np.random.rand(100)
print(A)

plt.hist(A, bins = 10, edgecolor = 'black', color = 'orange')
```

```
[0.38982101 0.17634333 0.19932874 0.05514393 0.85095229 0.78066521
 0.55291456 0.94185499 0.22925889 0.31284687 0.48597136 0.13292217
 0.10394975 0.86148781 0.23350566 0.27826535 0.54868531 0.4484815
 0.09336584 0.0658726  0.17532795 0.36966147 0.70012994 0.9355079
 0.11802387 0.06654189 0.98873953 0.18983474 0.31637963 0.57938179
 0.56472967 0.47296789 0.89071993 0.38210333 0.13862631 0.75821931
 0.69283556 0.35751399 0.89560215 0.10200244 0.1662713  0.6863203
 0.06438714 0.27239728 0.75907899 0.633203   0.88845873 0.84120378
 0.27807246 0.84827379 0.36540171 0.46977391 0.0199428  0.34088729
 0.46657438 0.19735985 0.53238118 0.48421323 0.51810924 0.45728068
 0.84159348 0.19594853 0.77571941 0.5420162  0.27308086 0.31620596
 0.95070264 0.54038031 0.09955353 0.96785555 0.28080902 0.93023387
 0.68177031 0.18047333 0.50374093 0.42909616 0.65127697 0.08432305
 0.35662103 0.09771217 0.79785628 0.01498219 0.28104937 0.76715636
 0.41651091 0.48967157 0.38211922 0.46914401 0.7106974  0.70460629
 0.76517999 0.01971633 0.87698185 0.42118743 0.29224446 0.53312554
 0.45357069 0.16386903 0.57916942 0.38671232]
```

Out[ ]:
```
(array([13., 12.,  9., 12., 13., 11.,  5.,  9.,  9.,  7.]),
 array([0.01498219, 0.11235793, 0.20973366, 0.30710939, 0.40448513,
        0.50186086, 0.59923659, 0.69661233, 0.79398806, 0.89136379,
        0.98873953]),
 <BarContainer object of 10 artists>)
```



In [ ]:
```python
# Bài 17: Viết chương trình tìm phần tử lớn thứ hai trong một mảng NumPy.
arr = np.array([1, 2,3 ,4, 5])
print(arr)

arr_sorted = np.unique(np.sort(arr))

print(arr_sorted[-2])
```

```
[1 2 3 4 5]
4
```

In [ ]:
```python
# Bài 18: Viết hàm kiểm tra một ma trận có khả nghịch hay không bằng NumPy.
def check_inv(mat):
  if (mat.shape[0] == mat.shape[1]):
    if (np.linalg.det(mat) != 0):
      return True

  return False

# Test case
# Tạo một ma trận khả nghịch (định thức khác 0)
matrix_invertible = np.array([[1, 2], [3, 4]])
print("Ma trận khả nghịch:\n", matrix_invertible)
print("Có khả nghịch không?", check_inv(matrix_invertible))

# Tạo một ma trận không khả nghịch (định thức bằng 0)
matrix_singular = np.array([[1, 2], [2, 4]])
print("\nMa trận không khả nghịch:\n", matrix_singular)
print("Có khả nghịch không?", check_inv(matrix_singular))

# Tạo một ma trận không vuông
matrix_non_square = np.array([[1, 2, 3], [4, 5, 6]])
print("\nMa trận không vuông:\n", matrix_non_square)
print("Có khả nghịch không?", check_inv(matrix_non_square))
```

```
Ma trận khả nghịch:
 [[1 2]
 [3 4]]
Có khả nghịch không? True

Ma trận không khả nghịch:
 [[1 2]
 [2 4]]
Có khả nghịch không? False

Ma trận không vuông:
 [[1 2 3]
 [4 5 6]]
Có khả nghịch không? False
```

In [ ]:
```
# Bài 19: Sinh mảng ngẫu nhiên 50 phần tử, sau đó tính giá trị trung vị và tứ ph
arr = np.random.rand(50)
print(arr)

med = np.median(arr)
print('Trung vị: ', med)

#np.percentile(data_array, q)
q_4 = [25, 50, 75]
q = np.percentile(arr, q_4)
print("Tứ phân vị: ", q)
```

```
[0.40307932 0.94506595 0.13499069 0.03309067 0.65791495 0.43392748
 0.11663962 0.30468401 0.66751229 0.60244322 0.25263482 0.43311768
 0.11490357 0.14408443 0.331803   0.26445276 0.93962662 0.38024843
 0.81412413 0.49311844 0.1706909  0.16724323 0.38355792 0.51786021
 0.11520155 0.31248805 0.22505734 0.43196906 0.31231764 0.83566918
 0.85287622 0.10262423 0.11670573 0.58569591 0.1689363  0.23157924
 0.86723566 0.97255613 0.80719108 0.21186808 0.78273775 0.80577313
 0.3111545  0.68210307 0.55378556 0.11618531 0.7439507  0.30865323
 0.8858677  0.88788387]
Trung vị:  0.39331861739453405
Tứ phân vị:  [0.2151654  0.39331862 0.7284888 ]
```

In [ ]:
```
# Bài 20: Viết chương trình mô phỏng ma trận hiệp phương sai từ dữ liệu ngẫu nhi
```

Bài số 17: Thư viện Pandas cơ bản

In [ ]:
```
import pandas as pd
```

In [ ]:
```
s = pd.Series([10, 20, 30, 40], index = ['a', 'b', 'c', 'd'])
print(s)
```

```
a    10
b    20
c    30
d    40
dtype: int64
```

In [ ]:
```
data = {'Tên': ['An', 'Bình', 'Cường'],
        'Tuổi': [20, 19, 21],
        'Điểm': [8.5, 9.0, 7.5]}
```

```
df= pd.DataFrame(data)
print(df)
```

```
    Tên  Tuổi  Điểm
0    An    20   8.5
1  Bình    19   9.0
2  Cường   21   7.5
```

In [ ]:
```
print(df.loc[0])
```

```
Tên      An
Tuổi     20
Điểm    8.5
Name: 0, dtype: object
```

In [ ]:
```
print(df.iloc[1])
```

```
Tên     Bình
Tuổi      19
Điểm     9.0
Name: 1, dtype: object
```

In [ ]:
```
print(df['Tên'])
```

```
0       An
1     Bình
2    Cường
Name: Tên, dtype: object
```

In [ ]:
```
df2 = pd.DataFrame({
    'A':[1, 2, None],
    'B':[4, None, 6]
})

print(df2.fillna(0))
```

```
    A    B
0  1.0  4.0
1  2.0  0.0
2  0.0  6.0
```

In [ ]:
```
print(df.sort_values(by='Điểm'))
print(df[df['Điểm'] > 8])
```

```
    Tên  Tuổi  Điểm
2  Cường   21   7.5
0    An    20   8.5
1  Bình    19   9.0
    Tên  Tuổi  Điểm
0    An    20   8.5
1  Bình    19   9.0
```

In [ ]:
```
print(df.describe())
```

```
       Tuổi      Điểm
count   3.0  3.000000
mean   20.0  8.333333
std     1.0  0.763763
min    19.0  7.500000
25%    19.5  8.000000
50%    20.0  8.500000
75%    20.5  8.750000
max    21.0  9.000000
```

In [ ]:
```
df3 = pd.DataFrame({
    'Lớp': ['A', 'A', 'B', 'B'],
    'Điểm': [8, 6, 9, 7]
})
print(df3.groupby('Lớp').median())
```

```
     Điểm
Lớp
A     7.0
B     8.0
```

In [ ]:
```
dfA = pd.DataFrame({'ID': [1, 2], 'Tên':['An', 'Bình']})
print(dfA)
dfB = pd.DataFrame({'ID':[1,2], 'Điểm':[8.5, 7.0]})
print(dfB)
merged = pd.merge(dfA, dfB, on = 'ID')
print(merged)
```

```
   ID   Tên
0   1    An
1   2  Bình
   ID  Điểm
0   1   8.5
1   2   7.0
   ID   Tên  Điểm
0   1    An   8.5
1   2  Bình   7.0
```

In [ ]:
```
dates = pd.date_range('2025-01-01', periods = 3)
print(dates)
ts = pd.Series([1, 2, 3], index = dates)
print(ts)
```

```
DatetimeIndex(['2025-01-01', '2025-01-02', '2025-01-03'], dtype='datetime64[ns]',
freq='D')
2025-01-01    1
2025-01-02    2
2025-01-03    3
Freq: D, dtype: int64
```

In [ ]:
```
print(df['Điểm'].apply(lambda x: x**2))
```

```
0    64
1    36
2    81
3    49
Name: Điểm, dtype: int64
```

Phần A

In [76]:
```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

In [78]:
```
working_path = "/content/drive/MyDrive/HCMUS_3/Py4ds"
file_name = "DataScience_salaries_2024.csv"
```

In [79]:
```
full_path = working_path + '/' + file_name
df = pd.read_csv(full_path)
```

In [80]:
```
print(df.head())
```

```
   work_year experience_level employment_type                   job_title  \
0       2021               MI              FT                Data Scientist
1       2021               MI              FT               BI Data Analyst
2       2020               MI              FT                Data Scientist
3       2021               MI              FT                   ML Engineer
4       2022               SE              FT  Lead Machine Learning Engineer

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50

  company_location company_size
0               CL            L
1               US            L
2               HU            L
3               JP            S
4               IN            L
```

In [81]:
```
# Bài 27: Sử dụng các hàm chuỗi: str.upper, str.lower.
print("--- job_title (chữ hoa) ---")
print(df['job_title'].str.upper().head())

print("\n--- job_title (chữ thường) ---")
print(df['job_title'].str.lower().head())
```

```
--- job_title (chữ hoa) ---
0                   DATA SCIENTIST
1                  BI DATA ANALYST
2                   DATA SCIENTIST
3                      ML ENGINEER
4    LEAD MACHINE LEARNING ENGINEER
Name: job_title, dtype: object

--- job_title (chữ thường) ---
0                   data scientist
1                  bi data analyst
2                   data scientist
3                      ml engineer
4    lead machine learning engineer
Name: job_title, dtype: object
```

In [82]:
```
# Bài 28: Tách một cột chuỗi thành nhiều cột bằng str.split.
# (Tách cột 'job_title', ví dụ 'Data Scier')
```

```
df_split = df['job_title'].str.split(' ', expand=True)
print(df_split.head())
```

```
      0          1         2         3
0  Data  Scientist      None      None
1    BI       Data   Analyst      None
2  Data  Scientist      None      None
3    ML   Engineer      None      None
4  Lead    Machine  Learning  Engineer
```

In [83]:
```
# Bài 29: Kết hợp chuỗi trong hai cột bằng str.cat.
# (Ghép 'company_location' và 'company_size', ngăn cách bởi ' - ')
location_size = df['company_location'].str.cat(df['company_size'], sep=' - ')
print(location_size.head())
```

```
0    CL - L
1    US - L
2    HU - L
3    JP - S
4    IN - L
Name: company_location, dtype: object
```

In [84]:
```
df['dummy_date_string'] = df['work_year'].astype(str) + '-01-15'
```

In [85]:
```
print(df.head(5))
```

```
   work_year experience_level employment_type                    job_title  \
0       2021               MI              FT                Data Scientist
1       2021               MI              FT               BI Data Analyst
2       2020               MI              FT                Data Scientist
3       2021               MI              FT                   ML Engineer
4       2022               SE              FT  Lead Machine Learning Engineer

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50

  company_location company_size dummy_date_string
0               CL            L        2021-01-15
1               US            L        2021-01-15
2               HU            L        2020-01-15
3               JP            S        2021-01-15
4               IN            L        2022-01-15
```

In [86]:
```
# Bài 30: Chuyển một cột ngày dạng chuỗi sang datetime.
# (Sử dụng cột 'dummy_date_string' đã tạo)
df['date_datetime'] = pd.to_datetime(df['dummy_date_string'])
print(df[['dummy_date_string', 'date_datetime']].head())
print("\nKiểm tra kiểu dữ liệu của cột mới:")
df.info()
```

```
  dummy_date_string date_datetime
0        2021-01-15    2021-01-15
1        2021-01-15    2021-01-15
2        2020-01-15    2020-01-15
3        2021-01-15    2021-01-15
4        2022-01-15    2022-01-15

Kiểm tra kiểu dữ liệu của cột mới:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14838 entries, 0 to 14837
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   work_year           14838 non-null  int64         
 1   experience_level    14838 non-null  object        
 2   employment_type     14838 non-null  object        
 3   job_title           14838 non-null  object        
 4   salary              14838 non-null  int64         
 5   salary_currency     14838 non-null  object        
 6   salary_in_usd       14838 non-null  int64         
 7   employee_residence  14838 non-null  object        
 8   remote_ratio        14838 non-null  int64         
 9   company_location    14838 non-null  object        
 10  company_size        14838 non-null  object        
 11  dummy_date_string   14838 non-null  object        
 12  date_datetime       14838 non-null  datetime64[ns]
dtypes: datetime64[ns](1), int64(4), object(8)
memory usage: 1.5+ MB
```

In [87]:
```
# Bài 31: Trích xuất năm từ cột thời gian bằng dt.year.
year_series = df['date_datetime'].dt.year
print(year_series.head())
```

```
0    2021
1    2021
2    2020
3    2021
4    2022
Name: date_datetime, dtype: int32
```

In [88]:
```
# Bài 32: Trích xuất tháng từ cột thời gian bằng dt.month.
month_series = df['date_datetime'].dt.month
print(month_series.head())
```

```
0    1
1    1
2    1
3    1
4    1
Name: date_datetime, dtype: int32
```

In [90]:
```
# Bài 33: Trích xuất ngày trong tuần bằng dt.dayofweek.
dayofweek_series = df['date_datetime'].dt.dayofweek
print(dayofweek_series.head())
```

```
0    4
1    4
2    2
3    4
4    5
Name: date_datetime, dtype: int32
```

In [91]:
```
# Bài 34: Tạo một cột mới bằng cách tính toán từ cột khác.
# (Ví dụ: Tính 10% thuế từ 'salary_in_usd')
df['tax_usd'] = df['salary_in_usd'] * 0.10
print(df[['salary_in_usd', 'tax_usd']].head())
```

```
   salary_in_usd  tax_usd
0          40038   4003.8
1          36259   3625.9
2          35735   3573.5
3          77364   7736.4
4          95386   9538.6
```

In [92]:
```
# Bài 35: Tính giá trị lớn nhất của một cột.
max_salary = df['salary_in_usd'].max()
print(f"Lương (USD) cao nhất: {max_salary}")
```

```
Lương (USD) cao nhất: 800000
```

In [93]:
```
# Bài 36: Tính giá trị nhỏ nhất của một cột.
min_salary = df['salary_in_usd'].min()
print(f"Lương (USD) thấp nhất: {min_salary}")
```

```
Lương (USD) thấp nhất: 15000
```

In [94]:
```
# Bài 37: Tính trung bình của một cột số.
mean_salary = df['salary_in_usd'].mean()
print(f"Lương (USD) trung bình: {mean_salary}")
```

```
Lương (USD) trung bình: 149874.71876263648
```

In [95]:
```
# Bài 38: Tính tổng giá trị của một cột số.
total_salary = df['salary_in_usd'].sum()
print(f"Tổng lương (USD) đã trả: {total_salary}")
```

```
Tổng lương (USD) đã trả: 2223841077
```

In [96]:
```
# Bài 39: Đếm số lượng phần tử khác nhau bằng nunique.
# (Có bao nhiêu 'job_title' khác nhau?)
unique_job_count = df['job_title'].nunique()
print(f"Số lượng job title khác nhau: {unique_job_count}")
```

```
Số lượng job title khác nhau: 153
```

In [97]:
```
# Bài 40: Thống kê tần suất các giá trị bằng value_counts.
# (Thống kê số lượng nhân viên theo 'experience_level')
exp_level_counts = df['experience_level'].value_counts()
print(exp_level_counts)
```

```
experience_level
SE    9696
MI    3553
EN    1148
EX     441
Name: count, dtype: int64
```

In [98]:
```
# Bài 41: Sắp xếp dữ liệu theo một cột tăng dần (ví dụ: salary_in_usd).
df_sorted_asc = df.sort_values(by='salary_in_usd')
print(df_sorted_asc.head())
```

```
       work_year experience_level employment_type  \
14835       2021               EN              FT
14836       2022               EN              FT
14833       2022               MI              FT
14834       2020               EX              FT
64          2023               MI              FT

                            job_title  salary salary_currency  salary_in_usd  \
14835       Machine Learning Developer   15000             USD          15000
14836                     Data Analyst   15000             USD          15000
14833  Business Intelligence Developer   15000             USD          15000
14834               Staff Data Analyst   15000             USD          15000
64                        Data Analyst  866000             PHP          15680

      employee_residence  remote_ratio company_location company_size  \
14835                 TH           100               TH            L
14836                 ID             0               ID            L
14833                 GH           100               GH            M
14834                 NG             0               CA            M
64                    PH            50               PH            L

      dummy_date_string date_datetime  tax_usd
14835        2021-01-15    2021-01-15   1500.0
14836        2022-01-15    2022-01-15   1500.0
14833        2022-01-15    2022-01-15   1500.0
14834        2020-01-15    2020-01-15   1500.0
64           2023-01-15    2023-01-15   1568.0
```

In [99]:
```
# Bài 42: Sắp xếp dữ liệu theo một cột giảm dần (ví dụ: salary_in_usd).
df_sorted_desc = df.sort_values(by='salary_in_usd', ascending=False)
print(df_sorted_desc.head())
```

```
    work_year experience_level employment_type                 job_title  \
66       2024               MI              FT               AI Architect
68       2024               EN              FT               Data Analyst
71       2024               SE              FT               Data Analyst
72       2024               MI              FT  Machine Learning Scientist
74       2023               MI              FT   Machine Learning Engineer

    salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
66  800000             USD         800000                 CA           100
68  774000             USD         774000                 MX             0
71  750000             USD         750000                 US             0
72  750000             USD         750000                 US             0
74  750000             USD         750000                 US             0

   company_location company_size dummy_date_string date_datetime  tax_usd
66               CA            M        2024-01-15    2024-01-15  80000.0
68               MX            M        2024-01-15    2024-01-15  77400.0
71               US            M        2024-01-15    2024-01-15  75000.0
72               US            M        2024-01-15    2024-01-15  75000.0
74               US            M        2023-01-15    2023-01-15  75000.0
```

In [100…
```
# Bài 43: Gộp nhóm theo một cột và tính trung bình (ví dụ: lương trung bình theo
mean_salary_by_exp = df.groupby('experience_level')['salary_in_usd'].mean()
print(mean_salary_by_exp)
```

```
experience_level
EN     91656.841463
EX    194730.210884
MI    125386.553054
SE    163700.967100
Name: salary_in_usd, dtype: float64
```

In [101…
```
# Bài 44: Gộp nhóm theo nhiều cột và tính tổng (ví dụ: tổng lương theo 'work_yea
sum_salary_by_group = df.groupby(['work_year', 'experience_level'])['salary_in_u
print(sum_salary_by_group)
```

```
work_year  experience_level
2020       EN                   1466654
           EX                    719833
           MI                   2847999
           SE                   2634329
2021       EN                   2919301
           EX                   1861280
           MI                   7553200
           SE                   9449231
2022       EN                   9282169
           EX                   7718672
           MI                  36778569
           SE                 168256117
2023       EN                  42301168
           EX                  46298035
           MI                 212477371
           SE                1008571996
2024       EN                  49252762
           EX                  29278203
           MI                 185841284
           SE                 398332904
Name: salary_in_usd, dtype: int64
```

In [102…
```
# Bài 45: Tính nhiều chỉ số thống kê trong groupby bằng agg.
# (Thống kê lương theo 'job_title': đếm, trung bình, lớn nhất)
stats_by_job = df.groupby('job_title')['salary_in_usd'].agg(['count', 'mean', 'm
# Sắp xếp theo 'count' để xem các job phổ biến nhất
print(stats_by_job.sort_values(by='count', ascending=False).head(10))
```

```
                           count           mean     max
job_title
Data Engineer               3162  146780.174257  750000
Data Scientist              3015  154179.810945  750000
Data Analyst                2189  108031.788945  774000
Machine Learning Engineer   1542  188014.814527  750000
Research Scientist           475  194217.117895  720000
Analytics Engineer           403  159616.397022  750000
Applied Scientist            383  190350.707572  414000
Data Architect               369  163499.723577  400000
Research Engineer            276  190154.568841  720000
Business Intelligence Engineer  230  140673.973913  259000
```

In [103…
```
# Bài 46: Tạo bảng tổng hợp bằng pivot_table (lương trung bình, index='experienc
pivot = df.pivot_table(values='salary_in_usd',
                       index='experience_level',
                       columns='work_year',
                       aggfunc='mean',
                       fill_value=0) # Điền 0 nếu không có dữ liệu
print(pivot)
```

```
work_year                  2020           2021           2022           2023  \
experience_level
EN                 69840.666667   63463.065217   80018.698276   91166.310345
EX                179958.250000  186128.000000  188260.292683  191314.194215
MI                 91870.935484   86818.390805  102733.432961  123821.311772
SE                138648.894737  125989.746667  147982.512753  165421.026078

work_year                  2024
experience_level
EN                 98308.906188
EX                203320.854167
MI                136547.600294
SE                168214.908784
```

In [104…

```python
import pandas as pd

# Tạo một DataFrame phụ (df_phu) để demo các bài 47-50
# df_mapping chứa thông tin về châu lục của một số quốc gia
data_map = {'company_location': ['US', 'IN', 'JP', 'DE', 'GB', 'FR', 'VN'],
            'Continent': ['North America', 'Asia', 'Asia', 'Europe', 'Europe', '
df_mapping = pd.DataFrame(data_map)
print("--- DataFrame Phụ (df_mapping) cho merge/join ---")
print(df_mapping.head())

# df_concat_demo để demo bài 47 (concat)
data_concat = {'work_year': [2025],
               'experience_level': ['EN'],
               'job_title': ['Data Intern'],
               'salary_in_usd': [25000],
               'company_location': ['VN']}
df_concat_demo = pd.DataFrame(data_concat)
print("\n--- DataFrame Phụ (df_concat_demo) cho concat ---")
print(df_concat_demo)
```

```
--- DataFrame Phụ (df_mapping) cho merge/join ---
  company_location      Continent
0               US  North America
1               IN           Asia
2               JP           Asia
3               DE         Europe
4               GB         Europe

--- DataFrame Phụ (df_concat_demo) cho concat ---
   work_year experience_level    job_title  salary_in_usd company_location
0       2025               EN  Data Intern          25000               VN
```

In [105…

```python
# Bài 47: Nối hai DataFrame bằng concat
df_concatenated = pd.concat([df, df_concat_demo], ignore_index=True)
print("--- 5 dòng cuối của DF sau khi concat (sẽ thấy Data Intern) ---")
print(df_concatenated.tail(5))
```

```
--- 5 dòng cuối của DF sau khi concat (sẽ thấy Data Intern) ---
       work_year experience_level employment_type                  job_title  \
14834       2020               EX              FT          Staff Data Analyst
14835       2021               EN              FT  Machine Learning Developer
14836       2022               EN              FT                Data Analyst
14837       2020               EN              PT                 ML Engineer
14838       2025               EN             NaN                 Data Intern

        salary salary_currency  salary_in_usd employee_residence  \
14834  15000.0             USD          15000                 NG
14835  15000.0             USD          15000                 TH
14836  15000.0             USD          15000                 ID
14837  14000.0             EUR          15966                 DE
14838      NaN             NaN          25000                NaN

       remote_ratio company_location company_size dummy_date_string  \
14834           0.0               CA            M        2020-01-15
14835         100.0               TH            L        2021-01-15
14836           0.0               ID            L        2022-01-15
14837         100.0               DE            S        2020-01-15
14838           NaN               VN          NaN               NaN

      date_datetime  tax_usd
14834    2020-01-15   1500.0
14835    2021-01-15   1500.0
14836    2022-01-15   1500.0
14837    2020-01-15   1596.6
14838           NaT      NaN
```

In [107…

```
# Bài 48: Gộp hai DataFrame theo cột chung bằng merge (mặc định là inner join).
df_merged_inner = pd.merge(df, df_mapping, on='company_location')
print(df_merged_inner.head())
```

```
   work_year experience_level employment_type                   job_title  \
0       2021               MI              FT                BI Data Analyst
1       2021               MI              FT                   ML Engineer
2       2022               SE              FT  Lead Machine Learning Engineer
3       2021               MI              FT                   ML Engineer
4       2021               SE              FT          Data Science Manager

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  11000000             HUF          36259                 HU            50
1   8500000             JPY          77364                 JP            50
2   7500000             INR          95386                 IN            50
3   7000000             JPY          63711                 JP            50
4   7000000             INR          94665                 IN            50

  company_location company_size dummy_date_string date_datetime  tax_usd  \
0               US            L        2021-01-15    2021-01-15   3625.9
1               JP            S        2021-01-15    2021-01-15   7736.4
2               IN            L        2022-01-15    2022-01-15   9538.6
3               JP            S        2021-01-15    2021-01-15   6371.1
4               IN            L        2021-01-15    2021-01-15   9466.5

       Continent
0  North America
1           Asia
2           Asia
3           Asia
4           Asia
```

In [ ]:

In [108…
```
# Bài 49: Thực hiện join kiểu left join (giữ tất cả từ 'df', thêm 'Continent' từ
df_left_join = pd.merge(df, df_mapping, on='company_location', how='left')
print(df_left_join.head())
# (Nếu quốc gia nào không có trong df_mapping, cột Continent sẽ là NaN)
```

```
  work_year experience_level employment_type                   job_title  \
0      2021              MI              FT                Data Scientist   
1      2021              MI              FT              BI Data Analyst   
2      2020              MI              FT                Data Scientist   
3      2021              MI              FT                   ML Engineer   
4      2022              SE              FT  Lead Machine Learning Engineer   

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100   
1  11000000             HUF          36259                 HU            50   
2  11000000             HUF          35735                 HU            50   
3   8500000             JPY          77364                 JP            50   
4   7500000             INR          95386                 IN            50   

  company_location company_size dummy_date_string date_datetime  tax_usd  \
0               CL            L        2021-01-15    2021-01-15   4003.8   
1               US            L        2021-01-15    2021-01-15   3625.9   
2               HU            L        2020-01-15    2020-01-15   3573.5   
3               JP            S        2021-01-15    2021-01-15   7736.4   
4               IN            L        2022-01-15    2022-01-15   9538.6   

       Continent  
0            NaN  
1  North America  
2            NaN  
3           Asia  
4           Asia  
```

In [109…
```
# Bài 50: Thực hiện join kiểu inner join (giống hệt Bài 48).
df_inner_join = pd.merge(df, df_mapping, on='company_location', how='inner')
print(df_inner_join.head())
```

```
   work_year experience_level employment_type                    job_title  \
0       2021               MI              FT                BI Data Analyst
1       2021               MI              FT                   ML Engineer
2       2022               SE              FT  Lead Machine Learning Engineer
3       2021               MI              FT                   ML Engineer
4       2021               SE              FT           Data Science Manager

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  11000000             HUF          36259                 HU            50
1   8500000             JPY          77364                 JP            50
2   7500000             INR          95386                 IN            50
3   7000000             JPY          63711                 JP            50
4   7000000             INR          94665                 IN            50

  company_location company_size dummy_date_string date_datetime  tax_usd  \
0               US            L        2021-01-15    2021-01-15   3625.9
1               JP            S        2021-01-15    2021-01-15   7736.4
2               IN            L        2022-01-15    2022-01-15   9538.6
3               JP            S        2021-01-15    2021-01-15   6371.1
4               IN            L        2021-01-15    2021-01-15   9466.5

      Continent
0  North America
1           Asia
2           Asia
3           Asia
4           Asia
```

In [110…

```python
# Bài 51: Xuất dữ liệu ra file CSV.
df.to_csv('DataScience_salaries_OUTPUT.csv', index=False, encoding='utf-8-sig')
print("Đã xuất file DataScience_salaries_OUTPUT.csv")
```

```
Đã xuất file DataScience_salaries_OUTPUT.csv
```

In [111…

```python
# Bài 52: Đọc dữ liệu từ file Excel.

# tạo file mẫu trước:
df.head(10).to_excel('salaries_sample.xlsx', index=False, sheet_name='Salaries')
print("Đã tạo file 'salaries_sample.xlsx' để đọc.")

# đọc file Excel
df_from_excel = pd.read_excel('salaries_sample.xlsx', sheet_name='Salaries')
print("\nNội dung đọc từ Excel:")
print(df_from_excel)
```

```
Đã tạo file 'salaries_sample.xlsx' để đọc.

Nội dung đọc từ Excel:
   work_year experience_level employment_type                    job_title  \
0       2021               MI              FT                 Data Scientist
1       2021               MI              FT                BI Data Analyst
2       2020               MI              FT                 Data Scientist
3       2021               MI              FT                    ML Engineer
4       2022               SE              FT  Lead Machine Learning Engineer
5       2021               MI              FT                    ML Engineer
6       2021               SE              FT           Data Science Manager
7       2022               EN              FT                 Data Scientist
8       2022               EX              FT       Head of Machine Learning
9       2022               EN              FT              Research Engineer

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50
5   7000000             JPY          63711                 JP            50
6   7000000             INR          94665                 IN            50
7   6600000             HUF          17684                 HU           100
8   6000000             INR          76309                 IN            50
9   5500000             JPY          41809                 JP            50

  company_location company_size dummy_date_string date_datetime  tax_usd
0               CL            L        2021-01-15    2021-01-15   4003.8
1               US            L        2021-01-15    2021-01-15   3625.9
2               HU            L        2020-01-15    2020-01-15   3573.5
3               JP            S        2021-01-15    2021-01-15   7736.4
4               IN            L        2022-01-15    2022-01-15   9538.6
5               JP            S        2021-01-15    2021-01-15   6371.1
6               IN            L        2021-01-15    2021-01-15   9466.5
7               HU            M        2022-01-15    2022-01-15   1768.4
8               IN            L        2022-01-15    2022-01-15   7630.9
9               JP            L        2022-01-15    2022-01-15   4180.9
```

In [112…

```python
# Bài 53: Xuất dữ liệu ra file Excel.

df.to_excel('DataScience_salaries_OUTPUT.xlsx', index=False, sheet_name='Data')
print("Đã xuất file DataScience_salaries_OUTPUT.xlsx")
```

```
Đã xuất file DataScience_salaries_OUTPUT.xlsx
```

In [114…

```python
# Bài 54: Đọc dữ liệu từ JSON.
df.head(10).to_json('salaries_sample.json', orient='records', indent=4, force_as
print("Đã tạo file 'salaries_sample.json' để đọc.")

# đọc file JSON (orient='records')
df_from_json = pd.read_json('salaries_sample.json', orient='records')
print("\nNội dung đọc từ JSON:")
print(df_from_json)
```

```
Đã tạo file 'salaries_sample.json' để đọc.

Nội dung đọc từ JSON:
   work_year experience_level employment_type                    job_title  \
0       2021               MI              FT                 Data Scientist
1       2021               MI              FT                BI Data Analyst
2       2020               MI              FT                 Data Scientist
3       2021               MI              FT                    ML Engineer
4       2022               SE              FT  Lead Machine Learning Engineer
5       2021               MI              FT                    ML Engineer
6       2021               SE              FT           Data Science Manager
7       2022               EN              FT                 Data Scientist
8       2022               EX              FT       Head of Machine Learning
9       2022               EN              FT              Research Engineer

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50
5   7000000             JPY          63711                 JP            50
6   7000000             INR          94665                 IN            50
7   6600000             HUF          17684                 HU           100
8   6000000             INR          76309                 IN            50
9   5500000             JPY          41809                 JP            50

  company_location company_size dummy_date_string  date_datetime  tax_usd
0               CL            L        2021-01-15  1610668800000   4003.8
1               US            L        2021-01-15  1610668800000   3625.9
2               HU            L        2020-01-15  1579046400000   3573.5
3               JP            S        2021-01-15  1610668800000   7736.4
4               IN            L        2022-01-15  1642204800000   9538.6
5               JP            S        2021-01-15  1610668800000   6371.1
6               IN            L        2021-01-15  1610668800000   9466.5
7               HU            M        2022-01-15  1642204800000   1768.4
8               IN            L        2022-01-15  1642204800000   7630.9
9               JP            L        2022-01-15  1642204800000   4180.9
```

In [115…

```python
# Bài 55: Xuất dữ liệu ra JSON.
# (Sử dụng force_ascii=False để giữ tiếng Việt nếu có)
df.to_json('DataScience_salaries_OUTPUT.json', orient='records', indent=4, force
print("Đã xuất file DataScience_salaries_OUTPUT.json")
```

```
Đã xuất file DataScience_salaries_OUTPUT.json
```

In [116…

```python
# Bài 56: Đặt lại chỉ số về 0,1,2... bằng reset_index.
# Tạo một df có index tùy chỉnh (ví dụ: 'job_title')
df_indexed = df.set_index('job_title')
print("--- DF gốc với index 'job_title' ---")
print(df_indexed.head())

# Thực hiện reset_index
df_reset = df_indexed.reset_index()
print("\n--- DF sau khi reset_index (cột 'job_title' quay lại) ---")
print(df_reset.head())
```

```
--- DF gốc với index 'job_title' ---
                             work_year experience_level employment_type  \
job_title
Data Scientist                     2021              MI              FT
BI Data Analyst                    2021              MI              FT
Data Scientist                     2020              MI              FT
ML Engineer                        2021              MI              FT
Lead Machine Learning Engineer     2022              SE              FT

                               salary salary_currency  salary_in_usd  \
job_title
Data Scientist                 30400000             CLP          40038
BI Data Analyst                11000000             HUF          36259
Data Scientist                 11000000             HUF          35735
ML Engineer                     8500000             JPY          77364
Lead Machine Learning Engineer  7500000             INR          95386

                              employee_residence  remote_ratio  \
job_title
Data Scientist                                CL           100
BI Data Analyst                               HU            50
Data Scientist                                HU            50
ML Engineer                                   JP            50
Lead Machine Learning Engineer                IN            50

                              company_location company_size  \
job_title
Data Scientist                              CL            L
BI Data Analyst                             US            L
Data Scientist                              HU            L
ML Engineer                                 JP            S
Lead Machine Learning Engineer              IN            L

                              dummy_date_string date_datetime  tax_usd
job_title
Data Scientist                       2021-01-15    2021-01-15   4003.8
BI Data Analyst                      2021-01-15    2021-01-15   3625.9
Data Scientist                       2020-01-15    2020-01-15   3573.5
ML Engineer                          2021-01-15    2021-01-15   7736.4
Lead Machine Learning Engineer       2022-01-15    2022-01-15   9538.6

--- DF sau khi reset_index (cột 'job_title' quay lại) ---
                      job_title  work_year experience_level employment_type  \
0                 Data Scientist       2021              MI              FT
1                BI Data Analyst       2021              MI              FT
2                 Data Scientist       2020              MI              FT
3                    ML Engineer       2021              MI              FT
4  Lead Machine Learning Engineer       2022              SE              FT

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50

  company_location company_size dummy_date_string date_datetime  tax_usd
0               CL            L       2021-01-15    2021-01-15   4003.8
1               US            L       2021-01-15    2021-01-15   3625.9
2               HU            L       2020-01-15    2020-01-15   3573.5
```

```
3                JP              S         2021-01-15    2021-01-15    7736.4
4                IN              L         2022-01-15    2022-01-15    9538.6
```

In [117…
```
# Bài 57: Đặt một cột làm chỉ số bằng set_index (ví dụ: cột 'job_title').
df_set_index = df.set_index('job_title')
print(df_set_index.head())
```

```
                                work_year experience_level employment_type  \
job_title
Data Scientist                       2021              MI              FT
BI Data Analyst                      2021              MI              FT
Data Scientist                       2020              MI              FT
ML Engineer                          2021              MI              FT
Lead Machine Learning Engineer       2022              SE              FT

                                  salary salary_currency  salary_in_usd  \
job_title
Data Scientist                  30400000             CLP          40038
BI Data Analyst                 11000000             HUF          36259
Data Scientist                  11000000             HUF          35735
ML Engineer                      8500000             JPY          77364
Lead Machine Learning Engineer   7500000             INR          95386

                                employee_residence  remote_ratio  \
job_title
Data Scientist                                  CL           100
BI Data Analyst                                 HU            50
Data Scientist                                  HU            50
ML Engineer                                     JP            50
Lead Machine Learning Engineer                  IN            50

                                company_location company_size  \
job_title
Data Scientist                                CL            L
BI Data Analyst                               US            L
Data Scientist                                HU            L
ML Engineer                                   JP            S
Lead Machine Learning Engineer                IN            L

                                dummy_date_string date_datetime  tax_usd
job_title
Data Scientist                         2021-01-15    2021-01-15   4003.8
BI Data Analyst                        2021-01-15    2021-01-15   3625.9
Data Scientist                         2020-01-15    2020-01-15   3573.5
ML Engineer                            2021-01-15    2021-01-15   7736.4
Lead Machine Learning Engineer         2022-01-15    2022-01-15   9538.6
```

In [118…
```
# Bài 58: Đổi tên nhiều cột cùng lúc bằng rename.
df_renamed = df.rename(columns={'salary_in_usd': 'Luong_USD',
                                'work_year': 'Nam_Lam_Viec',
                                'experience_level': 'Cap_Do_KN'})

print(df_renamed.head())
```

```
   Nam_Lam_Viec Cap_Do_KN employment_type                    job_title  \
0          2021        MI              FT                 Data Scientist
1          2021        MI              FT                BI Data Analyst
2          2020        MI              FT                 Data Scientist
3          2021        MI              FT                   ML Engineer
4          2022        SE              FT  Lead Machine Learning Engineer

     salary salary_currency  Luong_USD employee_residence  remote_ratio  \
0  30400000             CLP      40038                 CL           100
1  11000000             HUF      36259                 HU            50
2  11000000             HUF      35735                 HU            50
3   8500000             JPY      77364                 JP            50
4   7500000             INR      95386                 IN            50

  company_location company_size dummy_date_string date_datetime  tax_usd
0               CL            L        2021-01-15    2021-01-15   4003.8
1               US            L        2021-01-15    2021-01-15   3625.9
2               HU            L        2020-01-15    2020-01-15   3573.5
3               JP            S        2021-01-15    2021-01-15   7736.4
4               IN            L        2022-01-15    2022-01-15   9538.6
```

In [119…

```python
# Bài 59: Tạo bản sao DataFrame bằng copy (bản sao sâu).
df_copy = df.copy()

# Thay đổi bản sao
df_copy.loc[0, 'salary_in_usd'] = 9999999
print("--- Bản sao đã thay đổi ---")
print(df_copy.head())
print("\n--- Bản gốc không đổi ---")
print(df.head())
```

```
--- Bản sao đã thay đổi ---
   work_year experience_level employment_type                     job_title  \
0       2021               MI              FT                  Data Scientist
1       2021               MI              FT                 BI Data Analyst
2       2020               MI              FT                  Data Scientist
3       2021               MI              FT                     ML Engineer
4       2022               SE              FT  Lead Machine Learning Engineer

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP        9999999                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50

  company_location company_size dummy_date_string date_datetime  tax_usd
0               CL            L        2021-01-15    2021-01-15   4003.8
1               US            L        2021-01-15    2021-01-15   3625.9
2               HU            L        2020-01-15    2020-01-15   3573.5
3               JP            S        2021-01-15    2021-01-15   7736.4
4               IN            L        2022-01-15    2022-01-15   9538.6

--- Bản gốc không đổi ---
   work_year experience_level employment_type                     job_title  \
0       2021               MI              FT                  Data Scientist
1       2021               MI              FT                 BI Data Analyst
2       2020               MI              FT                  Data Scientist
3       2021               MI              FT                     ML Engineer
4       2022               SE              FT  Lead Machine Learning Engineer

     salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0  30400000             CLP          40038                 CL           100
1  11000000             HUF          36259                 HU            50
2  11000000             HUF          35735                 HU            50
3   8500000             JPY          77364                 JP            50
4   7500000             INR          95386                 IN            50

  company_location company_size dummy_date_string date_datetime  tax_usd
0               CL            L        2021-01-15    2021-01-15   4003.8
1               US            L        2021-01-15    2021-01-15   3625.9
2               HU            L        2020-01-15    2020-01-15   3573.5
3               JP            S        2021-01-15    2021-01-15   7736.4
4               IN            L        2022-01-15    2022-01-15   9538.6
```

In [120…

```
# Bài 60: Kiểm tra thông tin kiểu dữ liệu bằng info.
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14838 entries, 0 to 14837
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   work_year           14838 non-null  int64         
 1   experience_level    14838 non-null  object        
 2   employment_type     14838 non-null  object        
 3   job_title           14838 non-null  object        
 4   salary              14838 non-null  int64         
 5   salary_currency     14838 non-null  object        
 6   salary_in_usd       14838 non-null  int64         
 7   employee_residence  14838 non-null  object        
 8   remote_ratio        14838 non-null  int64         
 9   company_location    14838 non-null  object        
 10  company_size        14838 non-null  object        
 11  dummy_date_string   14838 non-null  object        
 12  date_datetime       14838 non-null  datetime64[ns]
 13  tax_usd             14838 non-null  float64       
dtypes: datetime64[ns](1), float64(1), int64(4), object(8)
memory usage: 1.6+ MB
None
```

In [121…

```
# Bài 61: Kiểm tra kích thước dữ liệu bằng shape (số hàng, số cột).
print(df.shape)
```

```
(14838, 14)
```

Phần B: Vận dụng (40 câu)

Nhóm 1: Dữ liệu Sinh viên / Học sinh

Gợi ý data: Tạo một DataFrame ban đầu (hoặc đọc từ file) gồm các cột: HoTen, Lop, GioiTinh, Tuoi, DiemToan, DiemVan.

In [28]:

```
import pandas as pd
import matplotlib.pyplot as plt

!pip install openpyxl -q

# Gợi ý data: HoTen, Lop, GioiTinh, Tuoi, DiemToan, DiemVan
data = {
    'HoTen': ['Nguyễn Văn An', 'Trần Thị Bình', 'Lê Văn Cường', 'Phạm Thị Định',
    'Lop': ['10A1', '10A2', '10A1', '10A2', '10A3', '10A1', '10A3', '10A2'],
    'GioiTinh': ['Nam', 'Nữ', 'Nam', 'Nữ', 'Nam', 'Nữ', 'Nam', 'Nữ'],
    'Tuoi': [16, 17, 16, 17, 18, 16, 18, 17],
    'DiemToan': [8.5, 9.0, 7.0, 9.5, 5.0, 7.5, 6.5, 8.0],
    'DiemVan': [7.0, 8.5, 6.5, 9.0, 4.5, 8.0, 5.5, 7.5]
}
df = pd.DataFrame(data)

print("--- Dữ liệu Sinh viên/Học sinh ban đầu ---")
print(df)
```

```
--- Dữ liệu Sinh viên/Học sinh ban đầu ---
            HoTen   Lop GioiTinh  Tuoi  DiemToan  DiemVan
0    Nguyễn Văn An  10A1      Nam    16       8.5      7.0
1    Trần Thị Bình  10A2       Nữ    17       9.0      8.5
2     Lê Văn Cường  10A1      Nam    16       7.0      6.5
3    Phạm Thị Định  10A2       Nữ    17       9.5      9.0
4  Hoàng Văn Giang  10A3      Nam    18       5.0      4.5
5       Mai Thị Hà  10A1       Nữ    16       7.5      8.0
6      Đặng Văn Em  10A3      Nam    18       6.5      5.5
7      Bùi Thị Lan  10A2       Nữ    17       8.0      7.5
```

In [29]:
```
# Bài 1: Tạo một DataFrame sinh viên gồm tên, tuổi, điểm toán, điểm văn. Tính đi
# (Chúng ta sẽ sử dụng df từ Cell 0 và thêm cột DiemTB)
df['DiemTB'] = (df['DiemToan'] + df['DiemVan']) / 2
print(df[['HoTen', 'Tuoi', 'DiemToan', 'DiemVan', 'DiemTB']])
```

```
            HoTen  Tuoi  DiemToan  DiemVan  DiemTB
0    Nguyễn Văn An    16       8.5      7.0    7.75
1    Trần Thị Bình    17       9.0      8.5    8.75
2     Lê Văn Cường    16       7.0      6.5    6.75
3    Phạm Thị Định    17       9.5      9.0    9.25
4  Hoàng Văn Giang    18       5.0      4.5    4.75
5       Mai Thị Hà    16       7.5      8.0    7.75
6      Đặng Văn Em    18       6.5      5.5    6.00
7      Bùi Thị Lan    17       8.0      7.5    7.75
```

In [30]:
```
# Bài 2: Lọc ra những sinh viên có điểm trung bình trên 7.
# (Giả định df đã có cột DiemTB từ Bài 1)
df_tren_7 = df[df['DiemTB'] > 7.0]
print(df_tren_7)
```

```
          HoTen   Lop GioiTinh  Tuoi  DiemToan  DiemVan  DiemTB
0  Nguyễn Văn An  10A1      Nam    16       8.5      7.0    7.75
1  Trần Thị Bình  10A2       Nữ    17       9.0      8.5    8.75
3  Phạm Thị Định  10A2       Nữ    17       9.5      9.0    9.25
5     Mai Thị Hà  10A1       Nữ    16       7.5      8.0    7.75
7    Bùi Thị Lan  10A2       Nữ    17       8.0      7.5    7.75
```

In [31]:
```
# Bài 3: Thêm cột xếp loại học lực dựa trên điểm trung bình.
# (Giả định df đã có cột DiemTB từ Bài 1)
bins = [0, 4.9, 6.4, 7.9, 10.0]
labels = ['Yếu', 'Trung Bình', 'Khá', 'Giỏi']
# (right=True nghĩa là (0, 4.9], (4.9, 6.4]...)
df['XepLoai'] = pd.cut(df['DiemTB'], bins=bins, labels=labels, right=True)
print(df)
```

```
            HoTen   Lop GioiTinh  Tuoi  DiemToan  DiemVan  DiemTB     XepLoai
0    Nguyễn Văn An  10A1      Nam    16       8.5      7.0    7.75         Khá
1    Trần Thị Bình  10A2       Nữ    17       9.0      8.5    8.75        Giỏi
2     Lê Văn Cường  10A1      Nam    16       7.0      6.5    6.75         Khá
3    Phạm Thị Định  10A2       Nữ    17       9.5      9.0    9.25        Giỏi
4  Hoàng Văn Giang  10A3      Nam    18       5.0      4.5    4.75         Yếu
5       Mai Thị Hà  10A1       Nữ    16       7.5      8.0    7.75         Khá
6      Đặng Văn Em  10A3      Nam    18       6.5      5.5    6.00  Trung Bình
7      Bùi Thị Lan  10A2       Nữ    17       8.0      7.5    7.75         Khá
```

In [32]:
```
# Bài 4: Với bảng học sinh (tên, giới tính, điểm), tính điểm trung bình theo giớ
# (Giả định df đã có cột DiemTB từ Bài 1)
tb_theo_gioitinh = df.groupby('GioiTinh')['DiemTB'].mean()
print(tb_theo_gioitinh)
```

```
GioiTinh
Nam    6.3125
Nữ     8.3750
Name: DiemTB, dtype: float64
```

In [33]:
```
# Bài 5: Tạo bảng tổng hợp điểm trung bình theo lớp và giới tính.
# (Giả định df đã có cột DiemTB từ Bài 1)
bang_tong_hop = df.pivot_table(values='DiemTB', index='Lop', columns='GioiTinh',
print(bang_tong_hop)
```

```
GioiTinh    Nam        Nữ
Lop
10A1      7.250  7.750000
10A2        NaN  8.583333
10A3      5.375       NaN
```

In [34]:
```
# Bài 6: Với bảng điểm, tìm học sinh có điểm cao nhất trong từng lớp.
# (Sử dụng DiemTB)
idx_max = df.groupby('Lop')['DiemTB'].idxmax()
hoc_sinh_cao_nhat = df.loc[idx_max]
print(hoc_sinh_cao_nhat)
```

```
          HoTen   Lop GioiTinh  Tuoi  DiemToan  DiemVan  DiemTB     XepLoai
0  Nguyễn Văn An  10A1      Nam    16       8.5      7.0    7.75         Khá
3  Phạm Thị Định  10A2       Nữ    17       9.5      9.0    9.25        Giỏi
6    Đặng Văn Em  10A3      Nam    18       6.5      5.5    6.00  Trung Bình
```

In [35]:
```
# Bài 7: Với bảng điểm, tìm học sinh có điểm thấp nhất trong từng lớp.
# (Sử dụng DiemTB)
idx_min = df.groupby('Lop')['DiemTB'].idxmin()
hoc_sinh_thap_nhat = df.loc[idx_min]
print(hoc_sinh_thap_nhat)
```

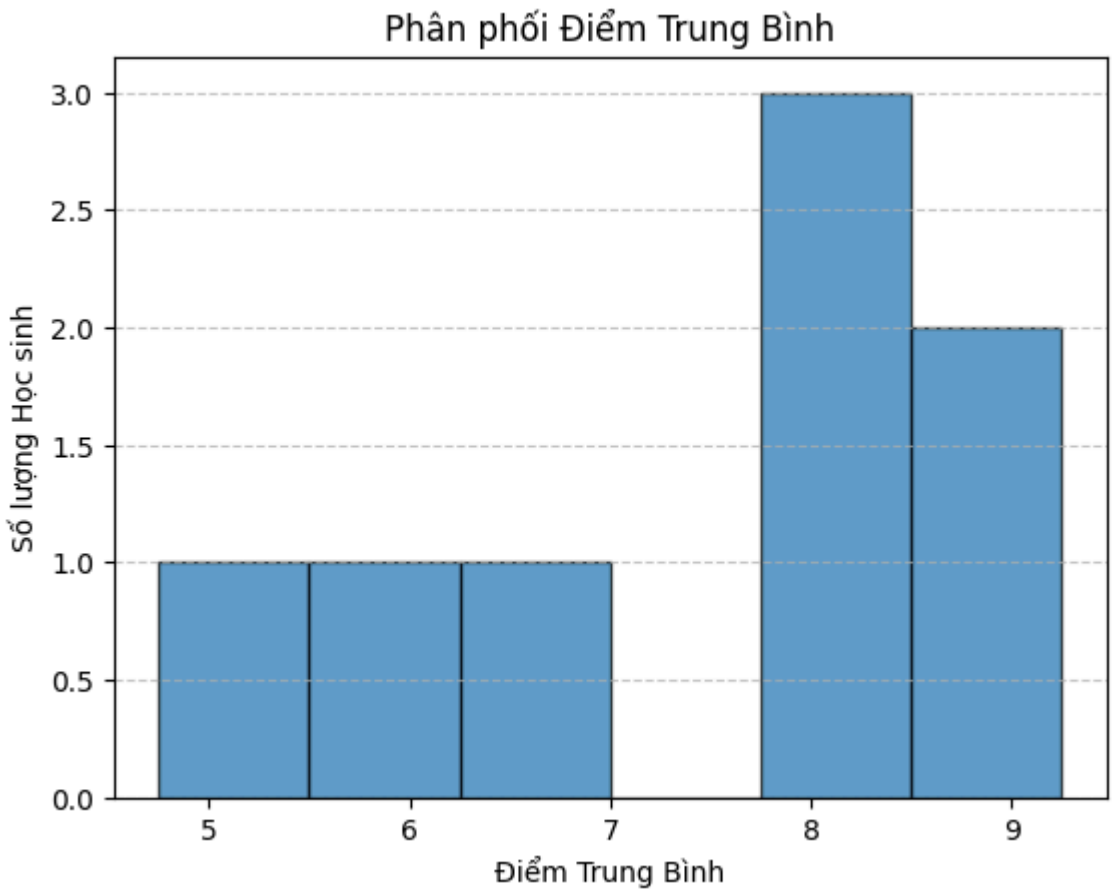
```
            HoTen   Lop GioiTinh  Tuoi  DiemToan  DiemVan  DiemTB XepLoai
2     Lê Văn Cường  10A1      Nam    16       7.0      6.5    6.75     Khá
7      Bùi Thị Lan  10A2       Nữ    17       8.0      7.5    7.75     Khá
4  Hoàng Văn Giang  10A3      Nam    18       5.0      4.5    4.75     Yếu
```

In [36]:
```
# Bài 8: Với bảng học sinh, tách cột họ tên thành hai cột: họ và tên.
# (Tách 1 lần từ bên phải (rsplit) để lấy 'Tên', phần còn lại là 'Họ và Đệm')
df_bai8 = df.copy()
df_bai8[['HoVaDem', 'Ten']] = df_bai8['HoTen'].str.rsplit(' ', n=1, expand=True)
print(df_bai8[['HoTen', 'HoVaDem', 'Ten']])
```

```
            HoTen     HoVaDem    Ten
0    Nguyễn Văn An  Nguyễn Văn     An
1    Trần Thị Bình    Trần Thị   Bình
2     Lê Văn Cường      Lê Văn  Cường
3    Phạm Thị Định    Phạm Thị   Định
4  Hoàng Văn Giang   Hoàng Văn  Giang
5       Mai Thị Hà     Mai Thị     Hà
6      Đặng Văn Em    Đặng Văn     Em
7      Bùi Thị Lan     Bùi Thị    Lan
```

In [37]:
```
# Bài 9: Với bảng dữ liệu điểm, vẽ biểu đồ phân phối điểm (sử dụng matplotlib).
# (Vẽ histogram của cột DiemTB, giả định đã chạy Bài 1)
plt.hist(df['DiemTB'], bins=6, edgecolor='black', alpha=0.7)
plt.title('Phân phối Điểm Trung Bình')
plt.xlabel('Điểm Trung Bình')
plt.ylabel('Số lượng Học sinh')
```

```
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



In [39]:
```
# Bài 10: Với bảng sinh viên, xuất dữ liệu ra CSV rồi đọc lại để kiểm tra.
file_name_csv = 'danh_sach_hoc_sinh.csv'
df.to_csv(file_name_csv, index=False, encoding='utf-8-sig')
print(f"Đã xuất file {file_name_csv}")

df_read_csv = pd.read_csv(file_name_csv)
print("\n--- Dữ liệu đọc lại từ CSV ---")
print(df_read_csv)
```

```
Đã xuất file danh_sach_hoc_sinh.csv

--- Dữ liệu đọc lại từ CSV ---
             HoTen   Lop GioiTinh  Tuoi  DiemToan  DiemVan  DiemTB     XepLoai
0    Nguyễn Văn An  10A1      Nam    16       8.5      7.0    7.75         Khá
1    Trần Thị Bình  10A2       Nữ    17       9.0      8.5    8.75        Giỏi
2     Lê Văn Cường  10A1      Nam    16       7.0      6.5    6.75         Khá
3    Phạm Thị Định  10A2       Nữ    17       9.5      9.0    9.25        Giỏi
4  Hoàng Văn Giang  10A3      Nam    18       5.0      4.5    4.75         Yếu
5       Mai Thị Hà  10A1       Nữ    16       7.5      8.0    7.75         Khá
6      Đặng Văn Em  10A3      Nam    18       6.5      5.5    6.00  Trung Bình
7      Bùi Thị Lan  10A2       Nữ    17       8.0      7.5    7.75         Khá
```

In [40]:
```
# Bài 11: Với bảng dữ liệu sinh viên, xuất dữ liệu ra Excel có nhiều sheet (mỗi
file_name_excel = 'hoc_sinh_theo_lop.xlsx'

with pd.ExcelWriter(file_name_excel, engine='openpyxl') as writer:
    # Lấy danh sách các lớp duy nhất (đã sắp xếp)
    all_lop = sorted(df['Lop'].unique())

    for lop in all_lop:
```

```
        # Lọc dữ liệu theo từng lớp
        df_lop = df[df['Lop'] == lop]
        # Xuất ra sheet có tên là lớp đó
        df_lop.to_excel(writer, sheet_name=lop, index=False)

print(f"Đã xuất file {file_name_excel} với các sheet: {all_lop}")
```

```
Đã xuất file hoc_sinh_theo_lop.xlsx với các sheet: ['10A1', '10A2', '10A3']
```

Nhóm 2: Dữ liệu Bán hàng / Sản phẩm

Gợi ý data: Tạo/đọc file DataFrame bán hàng gồm: NgayBan, MaSP, TenSP, SoLuong, DonGia. Có thể có một vài giá trị DonGia bị thiếu (NaN).

In [41]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Gợi ý data: NgayBan, MaSP, TenSP, SoLuong, DonGia (có NaN)
data = {
    'NgayBan': ['2025-10-01', '2025-10-01', '2025-10-02', '2025-10-03', '2025-11
    'MaSP': ['SP001', 'SP002', 'SP001', 'SP003', 'SP002', 'SP003', 'SP001', 'SP0
    'TenSP': ['Laptop', 'Mouse', 'Laptop', 'Keyboard', 'Mouse', 'Keyboard', 'Lap
    'SoLuong': [5, 10, 3, 7, 15, 8, 4, 12],
    'DonGia': [20000, 500, 20000, 700, 500, np.nan, 21000, 550] # Một giá trị Do
}
df = pd.DataFrame(data)

# Chuyển đổi cột NgayBan sang kiểu datetime
df['NgayBan'] = pd.to_datetime(df['NgayBan'])

print("--- Dữ liệu Bán hàng ban đầu ---")
print(df)
print("\nThông tin dữ liệu (kiểm tra NaN và kiểu dữ liệu):")
df.info()
```

```
--- Dữ liệu Bán hàng ban đầu ---
     NgayBan   MaSP     TenSP  SoLuong   DonGia
0 2025-10-01  SP001    Laptop        5  20000.0
1 2025-10-01  SP002     Mouse       10    500.0
2 2025-10-02  SP001    Laptop        3  20000.0
3 2025-10-03  SP003  Keyboard        7    700.0
4 2025-11-01  SP002     Mouse       15    500.0
5 2025-11-01  SP003  Keyboard        8      NaN
6 2025-11-02  SP001    Laptop        4  21000.0
7 2025-11-03  SP002     Mouse       12    550.0

Thông tin dữ liệu (kiểm tra NaN và kiểu dữ liệu):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   NgayBan  8 non-null      datetime64[ns]
 1   MaSP     8 non-null      object
 2   TenSP    8 non-null      object
 3   SoLuong  8 non-null      int64
 4   DonGia   7 non-null      float64
dtypes: datetime64[ns](1), float64(1), int64(1), object(2)
memory usage: 452.0+ bytes
```

In [42]:
```
# Bài 1: Với một bảng bán hàng (sản phẩm, ngày, số lượng, giá), tính doanh thu m
df['DoanhThu'] = df['SoLuong'] * df['DonGia']
print("--- Dữ liệu sau khi tính DoanhThu (lưu ý dòng có NaN) ---")
print(df)
```

```
--- Dữ liệu sau khi tính DoanhThu (lưu ý dòng có NaN) ---
     NgayBan   MaSP     TenSP  SoLuong   DonGia  DoanhThu
0 2025-10-01  SP001    Laptop        5  20000.0  100000.0
1 2025-10-01  SP002     Mouse       10    500.0    5000.0
2 2025-10-02  SP001    Laptop        3  20000.0   60000.0
3 2025-10-03  SP003  Keyboard        7    700.0    4900.0
4 2025-11-01  SP002     Mouse       15    500.0    7500.0
5 2025-11-01  SP003  Keyboard        8      NaN       NaN
6 2025-11-02  SP001    Laptop        4  21000.0   84000.0
7 2025-11-03  SP002     Mouse       12    550.0    6600.0
```

In [43]:
```
# Bài 2: Tính tổng doanh thu theo từng sản phẩm.
tong_doanh_thu_sp = df.groupby('TenSP')['DoanhThu'].sum()
print(tong_doanh_thu_sp)
```

```
TenSP
Keyboard      4900.0
Laptop      244000.0
Mouse        19100.0
Name: DoanhThu, dtype: float64
```

In [48]:
```
# Bài 3: Tính doanh thu trung bình mỗi ngày.
# (Giả định df đã có cột DoanhThu từ Bài 1)
doanh_thu_moi_ngay = df.groupby('NgayBan')['DoanhThu'].sum()
print("--- Tổng doanh thu mỗi ngày ---")
print(doanh_thu_moi_ngay)

doanh_thu_tb_ngay = doanh_thu_moi_ngay.mean()
print(f"\nDoanh thu trung bình mỗi ngày là: {doanh_thu_tb_ngay:.2f}")
```

```
--- Tổng doanh thu mỗi ngày ---
NgayBan
2025-10-01    105000.0
2025-10-02     60000.0
2025-10-03      4900.0
2025-11-01      7500.0
2025-11-02     84000.0
2025-11-03      6600.0
Name: DoanhThu, dtype: float64

Doanh thu trung bình mỗi ngày là: 44666.67
```

In [45]:
```python
# Bài 4: Lọc ra sản phẩm có doanh thu cao nhất.
# (Sử dụng kết quả từ Bài 2)
# tong_doanh_thu_sp = df.groupby('TenSP')['DoanhThu'].sum()
sp_cao_nhat = tong_doanh_thu_sp.idxmax()
doanh_thu_cao_nhat = tong_doanh_thu_sp.max()
print(f"Sản phẩm có doanh thu cao nhất là: {sp_cao_nhat} (Doanh thu: {doanh_thu_
```

```
Sản phẩm có doanh thu cao nhất là: Laptop (Doanh thu: 244000.0)
```

In [47]:
```python
# Bài 5: Lọc ra ngày có doanh thu thấp nhất.
# doanh_thu_moi_ngay = df.groupby('NgayBan')['DoanhThu'].sum()
ngay_thap_nhat = doanh_thu_moi_ngay.idxmin()
doanh_thu_thap_nhat = doanh_thu_moi_ngay.min()

print(f"Ngày có doanh thu thấp nhất là: {ngay_thap_nhat.strftime('%Y-%m-%d')} (D
```

```
Ngày có doanh thu thấp nhất là: 2025-10-03 (Doanh thu: 4900.0)
```

In [49]:
```python
# Bài 6: Với bảng bán hàng, tính tỷ lệ phần trăm doanh thu mỗi sản phẩm so với t
# (Sử dụng kết quả 'tong_doanh_thu_sp' từ Bài 2)
tong_doanh_thu_toan_bo = df['DoanhThu'].sum()
ty_le_doanh_thu = (tong_doanh_thu_sp / tong_doanh_thu_toan_bo) * 100
print("--- Tỷ lệ % doanh thu theo sản phẩm ---")
print(ty_le_doanh_thu.round(2).astype(str) + ' %')
```

```
--- Tỷ lệ % doanh thu theo sản phẩm ---
TenSP
Keyboard     1.83 %
Laptop      91.04 %
Mouse        7.13 %
Name: DoanhThu, dtype: object
```

In [50]:
```python
# Bài 7: Với bảng dữ liệu sản phẩm, thay giá trị thiếu trong cột giá bằng giá tr
df_bai7 = df.copy()
gia_trung_binh = df_bai7['DonGia'].mean()
df_bai7['DonGia'] = df_bai7['DonGia'].fillna(gia_trung_binh)

print(f"Giá trị trung bình để điền vào NaN: {gia_trung_binh:.2f}")
print("\n--- Bảng dữ liệu sau khi điền giá trị thiếu ---")
# Dòng có MaSP 'SP003' và NgayBan '2025-11-01' đã được cập nhật
print(df_bai7)
```

```
Giá trị trung bình để điền vào NaN: 9035.71

--- Bảng dữ liệu sau khi điền giá trị thiếu ---
     NgayBan   MaSP     TenSP  SoLuong        DonGia  DoanhThu
0 2025-10-01  SP001    Laptop        5  20000.000000  100000.0
1 2025-10-01  SP002     Mouse       10    500.000000    5000.0
2 2025-10-02  SP001    Laptop        3  20000.000000   60000.0
3 2025-10-03  SP003  Keyboard        7    700.000000    4900.0
4 2025-11-01  SP002     Mouse       15    500.000000    7500.0
5 2025-11-01  SP003  Keyboard        8   9035.714286       NaN
6 2025-11-02  SP001    Laptop        4  21000.000000   84000.0
7 2025-11-03  SP002     Mouse       12    550.000000    6600.0
```

In [51]:
```python
# Bài 8: Với bảng dữ liệu bán hàng, tạo bảng tổng hợp doanh thu theo sản phẩm và
# (Giả định df đã có cột DoanhThu từ Bài 1)
df['Thang'] = df['NgayBan'].dt.month
bang_tong_hop = df.pivot_table(values='DoanhThu',
                               index='TenSP',
                               columns='Thang',
                               aggfunc='sum',
                               fill_value=0) # fill_value=0 để điền 0 vào các th
print(bang_tong_hop)
```

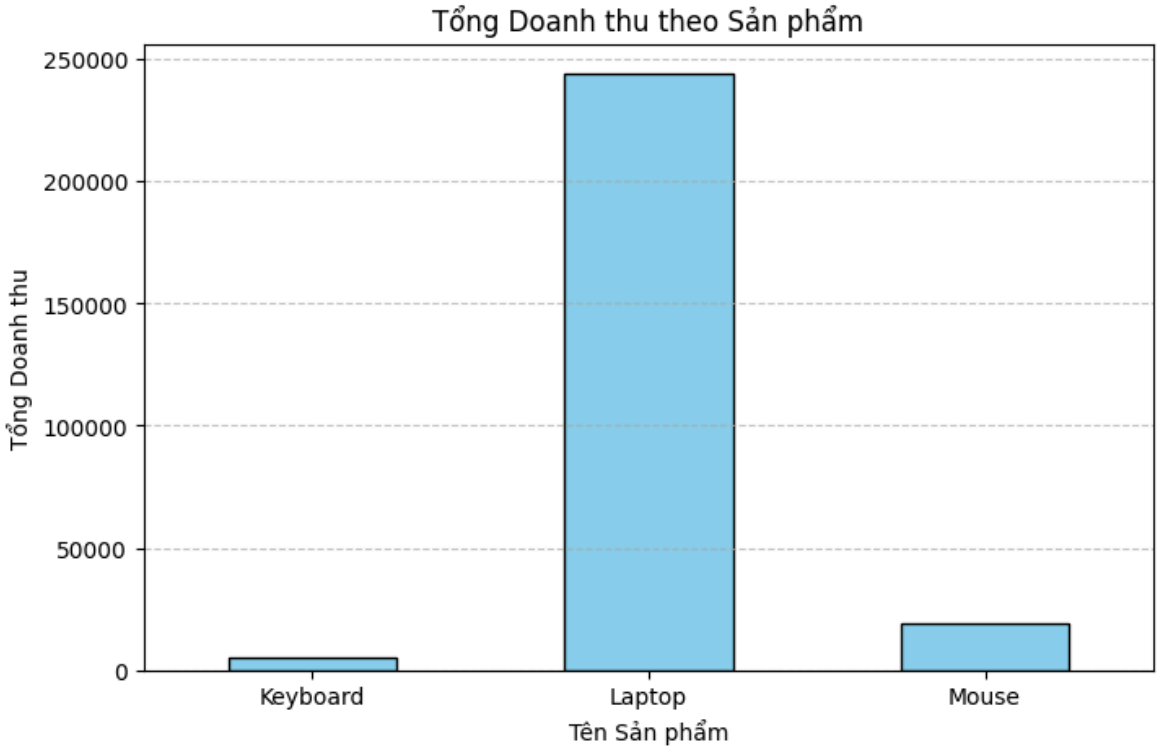
```
Thang           10       11
TenSP
Keyboard    4900.0      0.0
Laptop    160000.0  84000.0
Mouse       5000.0  14100.0
```

In [52]:
```python
# Bài 9: Với bảng dữ liệu bán hàng, tìm sản phẩm có tốc độ tăng trưởng doanh thu
# nhất theo tháng.
# (Sử dụng 'bang_tong_hop' từ Bài 8)

# Tính phần trăm thay đổi theo cột (axis=1), tức là theo tháng
# (DoanhThu_Thang11 - DoanhThu_Thang10) / DoanhThu_Thang10
tang_truong = bang_tong_hop.pct_change(axis='columns')

print("--- Bảng % tăng trưởng (NaN là tháng đầu tiên) ---")
print(tang_truong)

# Lấy cột của tháng cuối cùng (trong ví dụ này là tháng 11)
tang_truong_thang_cuoi = tang_truong.iloc[:, -1]
sp_tang_truong_nhanh_nhat = tang_truong_thang_cuoi.idxmax()
print(f"\nSản phẩm tăng trưởng nhanh nhất: {sp_tang_truong_nhanh_nhat} ({tang_tr
```

```
--- Bảng % tăng trưởng (NaN là tháng đầu tiên) ---
Thang     10     11
TenSP
Keyboard NaN -1.000
Laptop   NaN -0.475
Mouse    NaN  1.820

Sản phẩm tăng trưởng nhanh nhất: Mouse (182.00%)
```

In [53]:
```python
# Bài 10: Với bảng dữ liệu bán hàng, vẽ biểu đồ cột doanh thu theo sản phẩm.
# (Sử dụng 'tong_doanh_thu_sp' từ Bài 2)
tong_doanh_thu_sp.plot(kind='bar', figsize=(8, 5), color='skyblue', edgecolor='b
plt.title('Tổng Doanh thu theo Sản phẩm')
plt.ylabel('Tổng Doanh thu')
plt.xlabel('Tên Sản phẩm')
plt.xticks(rotation=0) # Xoay nhãn trục x
```

```
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



In [54]:
```
# Bài 11: Với bảng dữ liệu sản phẩm, lưu dữ liệu ra JSON và đọc lại.


file_name_json = 'danh_sach_ban_hang.json'
df.to_json(file_name_json, orient='records', indent=4, force_ascii=False, date_f
print(f"Đã xuất file {file_name_json}")

df_read_json = pd.read_json(file_name_json, orient='records')
df_read_json['NgayBan'] = pd.to_datetime(df_read_json['NgayBan'])

print("\n--- Dữ liệu đọc lại từ JSON ---")
print(df_read_json)
df_read_json.info()
```

```
Đã xuất file danh_sach_ban_hang.json

--- Dữ liệu đọc lại từ JSON ---
     NgayBan   MaSP     TenSP  SoLuong   DonGia  DoanhThu  Thang
0 2025-10-01  SP001    Laptop        5  20000.0  100000.0     10
1 2025-10-01  SP002     Mouse       10    500.0    5000.0     10
2 2025-10-02  SP001    Laptop        3  20000.0   60000.0     10
3 2025-10-03  SP003  Keyboard        7    700.0    4900.0     10
4 2025-11-01  SP002     Mouse       15    500.0    7500.0     11
5 2025-11-01  SP003  Keyboard        8      NaN       NaN     11
6 2025-11-02  SP001    Laptop        4  21000.0   84000.0     11
7 2025-11-03  SP002     Mouse       12    550.0    6600.0     11
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   NgayBan   8 non-null      datetime64[ns]
 1   MaSP      8 non-null      object
 2   TenSP     8 non-null      object
 3   SoLuong   8 non-null      int64
 4   DonGia    7 non-null      float64
 5   DoanhThu  7 non-null      float64
 6   Thang     8 non-null      int64
dtypes: datetime64[ns](1), float64(2), int64(2), object(2)
memory usage: 580.0+ bytes
```

Nhóm 3: Dữ liệu Nhân viên

Gợi ý data: Tạo DataFrame nhân viên gồm: TenNV, PhongBan, NamSinh, Luong.

In [55]:
```python
import pandas as pd
from datetime import datetime

# Gợi ý data: TenNV, PhongBan, NamSinh, Luong
data = {
    'TenNV': ['An', 'Bình', 'Cường', 'Định', 'Giang', 'Hà', 'Em', 'Lan', 'Minh',
    'PhongBan': ['Kế toán', 'Nhân sự', 'Kỹ thuật', 'Kỹ thuật', 'Nhân sự', 'Kế to
    'NamSinh': [1990, 1995, 1988, 1992, 1998, 1991, 1985, 2000, 1993, 1996],
    'Luong': [5000, 3500, 7000, 6500, 3800, 5200, 8000, 4000, 4500, 3600]
}
df = pd.DataFrame(data)

print("--- Dữ liệu Nhân viên ban đầu ---")
print(df)
```

```
--- Dữ liệu Nhân viên ban đầu ---
   TenNV   PhongBan  NamSinh  Luong
0     An    Kế toán     1990   5000
1   Bình    Nhân sự     1995   3500
2  Cường   Kỹ thuật     1988   7000
3   Định   Kỹ thuật     1992   6500
4  Giang    Nhân sự     1998   3800
5     Hà    Kế toán     1991   5200
6     Em   Kỹ thuật     1985   8000
7    Lan  Marketing     2000   4000
8   Minh  Marketing     1993   4500
9   Ngọc    Nhân sự     1996   3600
```

In [56]:
```
# Bài 1: Với bảng dữ liệu nhân viên (tên, phòng ban, lương), tính lương trung bì
luong_tb_pb = df.groupby('PhongBan')['Luong'].mean()
print(luong_tb_pb)
```

```
PhongBan
Kế toán      5100.000000
Kỹ thuật     7166.666667
Marketing    4250.000000
Nhân sự      3633.333333
Name: Luong, dtype: float64
```

In [57]:
```
# Bài 2: Xác định phòng ban có mức lương (trung bình) cao nhất.
# (Sử dụng kết quả từ Bài 1)
luong_tb_pb = df.groupby('PhongBan')['Luong'].mean()
pb_luong_cao_nhat = luong_tb_pb.idxmax()
luong_cao_nhat = luong_tb_pb.max()

print(f"Phòng ban có lương trung bình cao nhất: {pb_luong_cao_nhat} (Lương: {luo
```

```
Phòng ban có lương trung bình cao nhất: Kỹ thuật (Lương: 7166.67)
```

In [58]:
```
# Bài 3: Với bảng nhân viên, sắp xếp theo lương giảm dần và lấy 5 người cao nhất
top_5_luong = df.sort_values(by='Luong', ascending=False).head(5)
print(top_5_luong)
```

```
   TenNV  PhongBan  NamSinh  Luong
6     Em  Kỹ thuật     1985   8000
2  Cường  Kỹ thuật     1988   7000
3   Định  Kỹ thuật     1992   6500
5     Hà   Kế toán     1991   5200
0     An   Kế toán     1990   5000
```

In [59]:
```
# Bài 4: Với bảng nhân viên, thêm một cột tuổi tính từ cột năm sinh.
nam_hien_tai = datetime.now().year
df['Tuoi'] = nam_hien_tai - df['NamSinh']
print("--- Dữ liệu sau khi thêm cột Tuổi ---")
print(df)
```

```
--- Dữ liệu sau khi thêm cột Tuổi ---
   TenNV   PhongBan  NamSinh  Luong  Tuoi
0     An    Kế toán     1990   5000    35
1   Bình    Nhân sự     1995   3500    30
2  Cường   Kỹ thuật     1988   7000    37
3   Định   Kỹ thuật     1992   6500    33
4  Giang    Nhân sự     1998   3800    27
5     Hà    Kế toán     1991   5200    34
6     Em   Kỹ thuật     1985   8000    40
7    Lan  Marketing     2000   4000    25
8   Minh  Marketing     1993   4500    32
9   Ngọc    Nhân sự     1996   3600    29
```

In [60]:
```
# Bài 5: Với bảng dữ liệu nhân viên, lọc ra nhân viên có tuổi từ 30 đến 40 và lu
# (Giả định df đã có cột Tuoi từ Bài 4)
luong_trung_binh_chung = df['Luong'].mean()
print(f"Lương trung bình chung của công ty: {luong_trung_binh_chung:.2f}")

dieu_kien_tuoi = (df['Tuoi'] >= 30) & (df['Tuoi'] <= 40)
dieu_kien_luong = df['Luong'] > luong_trung_binh_chung

df_loc = df[dieu_kien_tuoi & dieu_kien_luong]
```

```
print("\n--- Nhân viên tuổi 30-40 và lương trên trung bình ---")
print(df_loc)
```

```
Lương trung bình chung của công ty: 5110.00

--- Nhân viên tuổi 30-40 và lương trên trung bình ---
   TenNV  PhongBan  NamSinh  Luong  Tuoi
2  Cường  Kỹ thuật     1988   7000    37
3   Định  Kỹ thuật     1992   6500    33
5     Hà   Kế toán     1991   5200    34
6     Em  Kỹ thuật     1985   8000    40
```

Nhóm 4: Dữ liệu Đơn hàng

Gợi ý data: Tạo DataFrame đơn hàng gồm: ID_DonHang, ID_KhachHang, NgayDat (dưới dạng chuỗi), TrangThai (vd: 'Đã giao', 'Đang xử lý', 'Đã hủy').

In [61]:
```
import pandas as pd

# Gợi ý data: ID_DonHang, ID_KhachHang, NgayDat (chuỗi), TrangThai
data_donhang = {
    'ID_DonHang': ['DH001', 'DH002', 'DH003', 'DH004', 'DH005', 'DH006', 'DH007'
    'ID_KhachHang': ['KH01', 'KH02', 'KH01', 'KH03', 'KH02', 'KH01', 'KH03'],
    'NgayDat': ['2025-11-01', '2025-11-01', '2025-11-02', '2025-11-03', '2025-11
    'TrangThai': ['Đã giao', 'Đang xử lý', 'Đã giao', 'Đã hủy', 'Đang xử lý', 'Đ
}
df = pd.DataFrame(data_donhang)

# Dữ liệu khách hàng (giả sử có) cho Bài 4
data_khachhang = {
    'ID_KhachHang': ['KH01', 'KH02', 'KH03'],
    'TenKH': ['Nguyễn An', 'Trần Bình', 'Lê Cường']
}
df_kh = pd.DataFrame(data_khachhang)

print("--- Dữ liệu Đơn hàng (df) ---")
print(df)
print("\n--- Dữ liệu Khách hàng (df_kh) ---")
print(df_kh)
```

```
--- Dữ liệu Đơn hàng (df) ---
  ID_DonHang ID_KhachHang     NgayDat   TrangThai
0      DH001         KH01  2025-11-01     Đã giao
1      DH002         KH02  2025-11-01  Đang xử lý
2      DH003         KH01  2025-11-02     Đã giao
3      DH004         KH03  2025-11-03      Đã hủy
4      DH005         KH02  2025-11-03  Đang xử lý
5      DH006         KH01  2025-11-03     Đã giao
6      DH007         KH03  2025-11-04  Đang xử lý

--- Dữ liệu Khách hàng (df_kh) ---
  ID_KhachHang      TenKH
0         KH01  Nguyễn An
1         KH02  Trần Bình
2         KH03   Lê Cường
```

In [62]:
```
# Bài 1: Với bảng dữ liệu đơn hàng (ID, ngày, trạng thái), đếm số đơn theo trạng
dem_theo_trang_thai = df['TrangThai'].value_counts()
```

```
print(dem_theo_trang_thai)
```

```
TrangThai
Đã giao       3
Đang xử lý    3
Đã hủy        1
Name: count, dtype: int64
```

In [63]:
```
# Bài 2: Tìm ngày có số đơn hàng nhiều nhất.
dem_theo_ngay = df['NgayDat'].value_counts()
ngay_nhieu_nhat = dem_theo_ngay.idxmax()
so_don_nhieu_nhat = dem_theo_ngay.max()

print(f"Ngày có số đơn hàng nhiều nhất: {ngay_nhieu_nhat} (với {so_don_nhieu_nha
```

```
Ngày có số đơn hàng nhiều nhất: 2025-11-03 (với 3 đơn)
```

In [64]:
```
# Bài 3: Với bảng dữ liệu đơn hàng, chuyển cột ngày từ chuỗi sang datetime và tr
df_bai3 = df.copy()
df_bai3['NgayDat_dt'] = pd.to_datetime(df_bai3['NgayDat'])
df_bai3['ThangDat'] = df_bai3['NgayDat_dt'].dt.month

print(df_bai3[['ID_DonHang', 'NgayDat_dt', 'ThangDat']])
print("\nThông tin kiểu dữ liệu (để xác nhận NgayDat_dt):")
df_bai3.info()
```

```
  ID_DonHang NgayDat_dt  ThangDat
0      DH001 2025-11-01        11
1      DH002 2025-11-01        11
2      DH003 2025-11-02        11
3      DH004 2025-11-03        11
4      DH005 2025-11-03        11
5      DH006 2025-11-03        11
6      DH007 2025-11-04        11

Thông tin kiểu dữ liệu (để xác nhận NgayDat_dt):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   ID_DonHang    7 non-null      object
 1   ID_KhachHang  7 non-null      object
 2   NgayDat       7 non-null      object
 3   TrangThai     7 non-null      object
 4   NgayDat_dt    7 non-null      datetime64[ns]
 5   ThangDat      7 non-null      int32
dtypes: datetime64[ns](1), int32(1), object(4)
memory usage: 440.0+ bytes
```

In [65]:
```
# Bài 4: Với bảng đơn hàng, ghép tên khách hàng (giả sử có) và ID đơn hàng thành
df_merged = pd.merge(df, df_kh, on='ID_KhachHang', how='left')
print("--- Bảng sau khi gộp Tên KH ---")
print(df_merged)

df_merged['ThongTinGhep'] = df_merged['TenKH'] + ' - ' + df_merged['ID_DonHang']
print("\n--- Bảng sau khi ghép chuỗi ---")
print(df_merged[['ID_DonHang', 'TenKH', 'ThongTinGhep']])
```

```
--- Bảng sau khi gộp Tên KH ---
  ID_DonHang ID_KhachHang     NgayDat   TrangThai      TenKH
0      DH001         KH01  2025-11-01     Đã giao  Nguyễn An
1      DH002         KH02  2025-11-01  Đang xử lý  Trần Bình
2      DH003         KH01  2025-11-02     Đã giao  Nguyễn An
3      DH004         KH03  2025-11-03      Đã hủy   Lê Cường
4      DH005         KH02  2025-11-03  Đang xử lý  Trần Bình
5      DH006         KH01  2025-11-03     Đã giao  Nguyễn An
6      DH007         KH03  2025-11-04  Đang xử lý   Lê Cường

--- Bảng sau khi ghép chuỗi ---
  ID_DonHang      TenKH       ThongTinGhep
0      DH001  Nguyễn An  Nguyễn An - DH001
1      DH002  Trần Bình  Trần Bình - DH002
2      DH003  Nguyễn An  Nguyễn An - DH003
3      DH004   Lê Cường   Lê Cường - DH004
4      DH005  Trần Bình  Trần Bình - DH005
5      DH006  Nguyễn An  Nguyễn An - DH006
6      DH007   Lê Cường   Lê Cường - DH007
```

Nhóm 5: Dữ liệu Khách hàng

Gợi ý data: Tạo DataFrame khách hàng gồm: ID_KhachHang, TenKH (viết hoa/thường lẫn lộn), Tuoi, TongChiTieu.

In [66]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Gợi ý data: ID_KhachHang, TenKH (Lẫn lộn hoa/thường), Tuoi, TongChiTieu
data = {
    'ID_KhachHang': ['KH01', 'KH02', 'KH03', 'KH04', 'KH05', 'KH06', 'KH07'],
    'TenKH': ['Nguyễn Văn A', 'trần thị B', 'LÊ VĂN C', 'Phạm Thị D', 'hoàng E',
    'Tuoi': [25, 40, 17, 33, 50, 29, 35],
    'TongChiTieu': [5000, 8000, 1500, 6000, 9000, 7500, 6200]
}
df = pd.DataFrame(data)

print("--- Dữ liệu Khách hàng ban đầu ---")
print(df)
```

```
--- Dữ liệu Khách hàng ban đầu ---
  ID_KhachHang         TenKH  Tuoi  TongChiTieu
0         KH01  Nguyễn Văn A    25         5000
1         KH02    trần thị B    40         8000
2         KH03      LÊ VĂN C    17         1500
3         KH04    Phạm Thị D    33         6000
4         KH05       hoàng E    50         9000
5         KH06          Đỗ F    29         7500
6         KH07          Vũ G    35         6200
```

In [67]:
```
# Bài 1: Với bảng dữ liệu khách hàng (tên, tuổi, chi tiêu), phân nhóm tuổi (dưới
# Phân nhóm: (0, 17] -> Dưới 18; (17, 35] -> 18-35; (35, inf) -> Trên 35
bins = [0, 17, 35, float('inf')]
labels = ['Dưới 18', '18-35', 'Trên 35']
df['NhomTuoi'] = pd.cut(df['Tuoi'], bins=bins, labels=labels, right=True)

print("--- Dữ liệu sau khi phân nhóm tuổi ---")
print(df)
```

```
# Tính chi tiêu trung bình theo nhóm tuổi
chi_tieu_tb_nhom_tuoi = df.groupby('NhomTuoi')['TongChiTieu'].mean()
print("\n--- Chi tiêu trung bình theo nhóm tuổi ---")
print(chi_tieu_tb_nhom_tuoi)
```

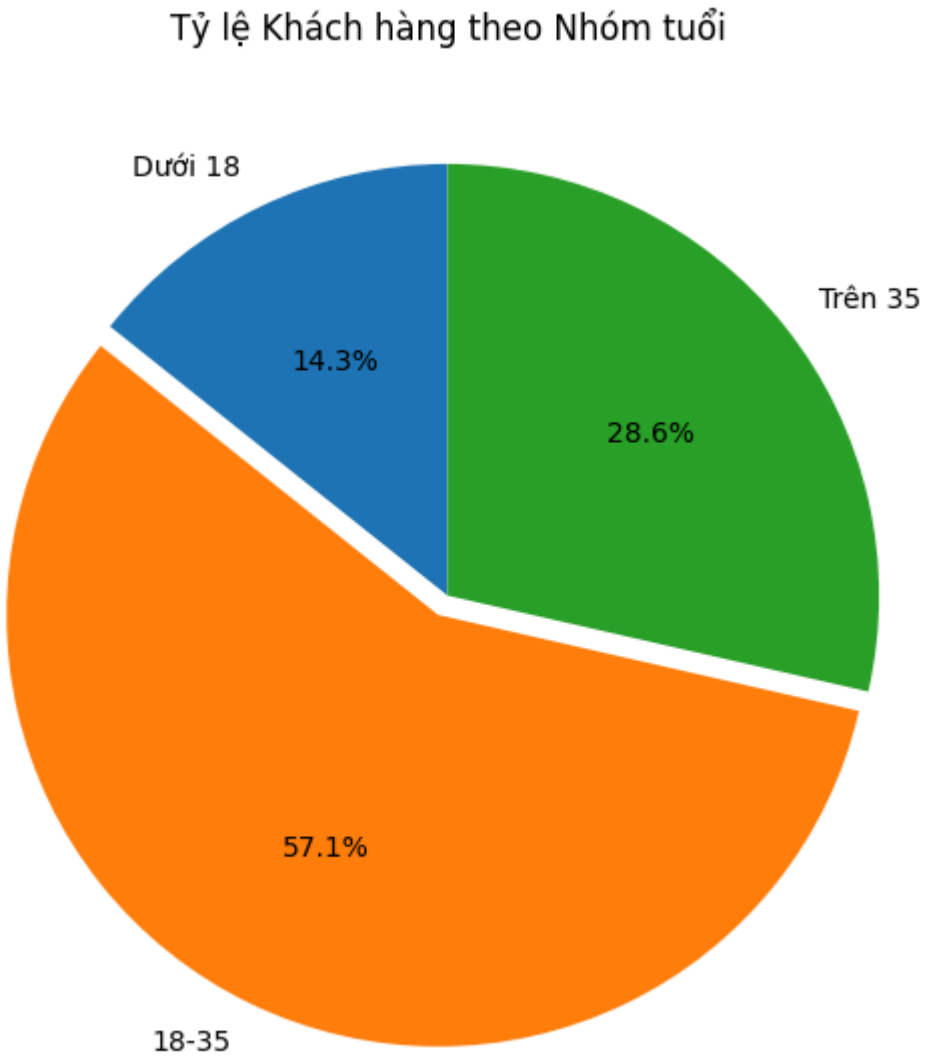
```
--- Dữ liệu sau khi phân nhóm tuổi ---
  ID_KhachHang          TenKH  Tuoi  TongChiTieu NhomTuoi
0         KH01  Nguyễn Văn A    25         5000    18-35
1         KH02    trần thị B    40         8000  Trên 35
2         KH03      LÊ VĂN C    17         1500  Dưới 18
3         KH04    Phạm Thị D    33         6000    18-35
4         KH05       hoàng E    50         9000  Trên 35
5         KH06          Đỗ F    29         7500    18-35
6         KH07          Vũ G    35         6200    18-35

--- Chi tiêu trung bình theo nhóm tuổi ---
NhomTuoi
Dưới 18    1500.0
18-35      6175.0
Trên 35    8500.0
Name: TongChiTieu, dtype: float64
/tmp/ipython-input-791693027.py:11: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass obs
erved=False to retain current behavior or observed=True to adopt the future defau
lt and silence this warning.
  chi_tieu_tb_nhom_tuoi = df.groupby('NhomTuoi')['TongChiTieu'].mean()
```

In [68]:
```
# Bài 2: Với bảng dữ liệu khách hàng, chuẩn hóa cột tên (chuyển hết về chữ hoa).
df_bai2 = df.copy()
df_bai2['TenKH_ChuanHoa'] = df_bai2['TenKH'].str.upper()
print(df_bai2[['TenKH', 'TenKH_ChuanHoa']])
```

```
          TenKH TenKH_ChuanHoa
0  Nguyễn Văn A   NGUYỄN VĂN A
1    trần thị B     TRẦN THỊ B
2      LÊ VĂN C       LÊ VĂN C
3    Phạm Thị D     PHẠM THỊ D
4       hoàng E        HOÀNG E
5          Đỗ F           ĐỖ F
6          Vũ G           VŨ G
```

In [69]:
```
# Bài 3: Với bảng dữ liệu khách hàng, đếm số khách theo từng nhóm tuổi và vẽ biể
# (Giả định df đã có cột 'NhomTuoi' từ Bài 1)
so_khach_theo_nhom_tuoi = df['NhomTuoi'].value_counts().sort_index()
print("--- Số khách theo nhóm tuổi ---")
print(so_khach_theo_nhom_tuoi)

# Vẽ biểu đồ tròn
so_khach_theo_nhom_tuoi.plot(kind='pie',
                             autopct='%1.1f%%',
                             figsize=(7, 7),
                             title='Tỷ lệ Khách hàng theo Nhóm tuổi',
                             startangle=90,
                             explode=(0, 0.05, 0)) # Tách nhẹ miếng '18-35'
plt.ylabel('')
plt.show()
```

```
--- Số khách theo nhóm tuổi ---
NhomTuoi
Dưới 18    1
18-35      4
Trên 35    2
Name: count, dtype: int64
```



Nhóm 6: Dữ liệu Thời tiết

Gợi ý data: Tạo DataFrame thời tiết gồm: Ngay (dạng datetime), NhietDoCao, NhietDoThap.

In [70]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Gợi ý data: Ngay (datetime), NhietDoCao, NhietDoThap
# Tạo dữ liệu cho 2 tuần, vắt qua 2 tháng (Tháng 10 và 11)
data = {
    'Ngay': pd.to_datetime(['2025-10-25', '2025-10-26', '2025-10-27', '2025-10-2
                            '2025-11-01', '2025-11-02', '2025-11-03', '2025-11-04
    'NhietDoCao': [28, 29, 30, 31, 30, 29, 31, 32, 33, 31, 30, 29, 28, 27],
    'NhietDoThap': [20, 21, 22, 23, 22, 21, 22, 24, 25, 24, 23, 22, 21, 20]
}
```

```
df = pd.DataFrame(data)

# Đặt cột 'Ngay' làm chỉ số (index)
df = df.set_index('Ngay')

print("--- Dữ liệu Thời tiết ban đầu ---")
print(df)
df.info()
```

```
--- Dữ liệu Thời tiết ban đầu ---
            NhietDoCao  NhietDoThap
Ngay
2025-10-25          28           20
2025-10-26          29           21
2025-10-27          30           22
2025-10-28          31           23
2025-10-29          30           22
2025-10-30          29           21
2025-10-31          31           22
2025-11-01          32           24
2025-11-02          33           25
2025-11-03          31           24
2025-11-04          30           23
2025-11-05          29           22
2025-11-06          28           21
2025-11-07          27           20
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14 entries, 2025-10-25 to 2025-11-07
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   NhietDoCao   14 non-null     int64
 1   NhietDoThap  14 non-null     int64
dtypes: int64(2)
memory usage: 336.0 bytes
```

In [71]:
```
# Bài 1: Với bảng dữ liệu thời tiết (ngày, nhiệt độ), tính nhiệt độ cao nhất the
# (df.index.month sẽ lấy tháng từ index (là cột Ngay))
nhiet_do_cao_nhat_thang = df.groupby(df.index.month)['NhietDoCao'].max()
nhiet_do_cao_nhat_thang.index.name = 'Thang'
print(nhiet_do_cao_nhat_thang)
```

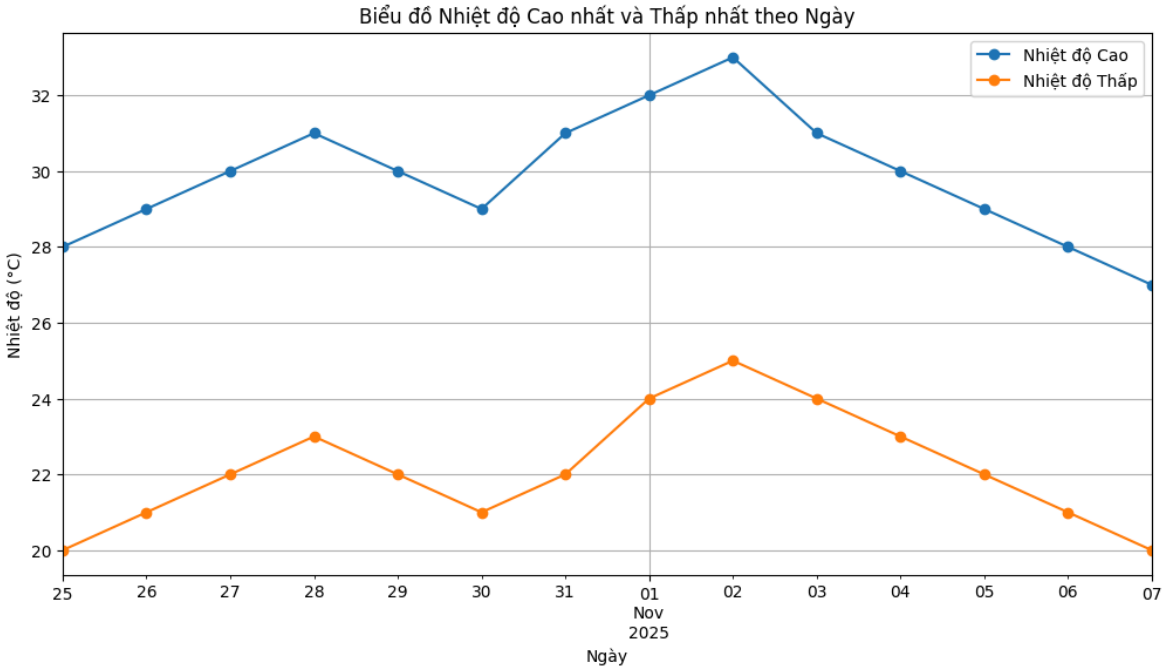
```
Thang
10    31
11    33
Name: NhietDoCao, dtype: int64
```

In [72]:
```
# Bài 2: Với bảng dữ liệu thời tiết, tính nhiệt độ trung bình theo ngày trong tu
# Tạo cột 'NhietDoTrungBinh'
df_bai2 = df.copy()
df_bai2['NhietDoTrungBinh'] = (df_bai2['NhietDoCao'] + df_bai2['NhietDoThap']) /

# Lấy tên ngày trong tuần từ index (là cột Ngay)
df_bai2['NgayTrongTuan'] = df_bai2.index.day_name()

# Tính trung bình của 'NhietDoTrungBinh' theo 'NgayTrongTuan'
tb_theo_ngay_tuan = df_bai2.groupby('NgayTrongTuan')['NhietDoTrungBinh'].mean()

# Sắp xếp lại theo thứ tự ngày trong tuần
```

```
days_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'
print(tb_theo_ngay_tuan.reindex(days_order))
```

```
NgayTrongTuan
Monday       26.75
Tuesday      26.75
Wednesday    25.75
Thursday     24.75
Friday       25.00
Saturday     26.00
Sunday       27.00
Name: NhietDoTrungBinh, dtype: float64
```

In [73]:
```
# Bài 3: Với bảng dữ liệu thời tiết, vẽ biểu đồ đường nhiệt độ theo ngày.
df[['NhietDoCao', 'NhietDoThap']].plot(kind='line',
                                       figsize=(12, 6),
                                       marker='o',
                                       linestyle='-')
plt.title('Biểu đồ Nhiệt độ Cao nhất và Thấp nhất theo Ngày')
plt.xlabel('Ngày')
plt.ylabel('Nhiệt độ (°C)')
plt.legend(['Nhiệt độ Cao', 'Nhiệt độ Thấp'])
plt.grid(True)
plt.show()
```



Nhóm 7: Bài tập Nối (Join) DataFrames

Gợi ý: Sử dụng các DataFrame đã tạo ở các nhóm trên (Khách hàng, Đơn hàng, Sản phẩm) và tạo thêm bảng DataFrame kho hàng.

In [74]:
```
import pandas as pd

# Data Khách hàng (từ Nhóm 5)
data_kh = {
    'ID_KhachHang': ['KH01', 'KH02', 'KH03', 'KH04'],
    'TenKH': ['Nguyễn An', 'Trần Bình', 'Lê Cường', 'Phạm Định'],
    'Tuoi': [25, 40, 17, 33],
```

```
    'TongChiTieu': [5000, 8000, 1500, 6000]
}
df_khachhang = pd.DataFrame(data_kh)

# Data Đơn hàng (từ Nhóm 4)
data_donhang = {
    'ID_DonHang': ['DH001', 'DH002', 'DH003', 'DH004', 'DH005'],
    'ID_KhachHang': ['KH01', 'KH02', 'KH01', 'KH03', 'KH02'], # KH04 chưa có đơn
    'NgayDat': ['2025-11-01', '2025-11-01', '2025-11-02', '2025-11-03', '2025-11
    'TrangThai': ['Đã giao', 'Đang xử lý', 'Đã giao', 'Đã hủy', 'Đang xử lý']
}
df_donhang = pd.DataFrame(data_donhang)

# Data Sản phẩm (từ Nhóm 2 - Rút gọn)
data_sp = {
    'MaSP': ['SP001', 'SP002', 'SP003', 'SP004'],
    'TenSP': ['Laptop', 'Mouse', 'Keyboard', 'Monitor']
}
df_sanpham = pd.DataFrame(data_sp)

# Data Kho hàng (Mới)
data_kho = {
    'MaSP': ['SP001', 'SP002', 'SP003'], # SP004 (Monitor) sẽ bị thiếu
    'SoLuongTon': [50, 200, 150]
}
df_khohang = pd.DataFrame(data_kho)

print("--- DF Khách hàng (Nhóm 5) ---")
print(df_khachhang)
print("\n--- DF Đơn hàng (Nhóm 4) ---")
print(df_donhang)
print("\n--- DF Sản phẩm (Nhóm 2) ---")
print(df_sanpham)
print("\n--- DF Kho hàng (Mới) ---")
print(df_khohang)
```

```
--- DF Khách hàng (Nhóm 5) ---
  ID_KhachHang      TenKH  Tuoi  TongChiTieu
0         KH01  Nguyễn An    25         5000
1         KH02  Trần Bình    40         8000
2         KH03   Lê Cường    17         1500
3         KH04  Phạm Định    33         6000

--- DF Đơn hàng (Nhóm 4) ---
  ID_DonHang ID_KhachHang     NgayDat   TrangThai
0      DH001         KH01  2025-11-01     Đã giao
1      DH002         KH02  2025-11-01  Đang xử lý
2      DH003         KH01  2025-11-02     Đã giao
3      DH004         KH03  2025-11-03      Đã hủy
4      DH005         KH02  2025-11-03  Đang xử lý

--- DF Sản phẩm (Nhóm 2) ---
    MaSP     TenSP
0  SP001    Laptop
1  SP002     Mouse
2  SP003  Keyboard
3  SP004   Monitor

--- DF Kho hàng (Mới) ---
    MaSP  SoLuongTon
0  SP001          50
1  SP002         200
2  SP003         150
```

In [75]:
```
# Bài 1: Nối bảng khách hàng (Nhóm 5) và bảng đơn hàng (Nhóm 4) để tìm chi tiêu
# từng khách hàng (và đếm số đơn hàng của họ).

# Bước 1: Đếm số đơn hàng của mỗi khách hàng từ bảng đơn hàng
so_don_moi_kh = df_donhang.groupby('ID_KhachHang').size().reset_index(name='SoDo
print("--- Số đơn mỗi khách (từ bảng Đơn hàng) ---")
print(so_don_moi_kh)

# Bước 2: Nối (merge) kết quả đếm với bảng khách hàng (để xem 'TongChiTieu' và '
# how='left' để giữ lại cả khách hàng KH04 (chưa có đơn)
df_ketqua_bai1 = pd.merge(df_khachhang, so_don_moi_kh, on='ID_KhachHang', how='l

# Thay thế NaN (khách chưa có đơn) bằng 0
df_ketqua_bai1['SoDonDaDat'] = df_ketqua_bai1['SoDonDaDat'].fillna(0)

print("\n--- Bảng tổng hợp chi tiêu và số đơn của khách hàng ---")
print(df_ketqua_bai1)
```

```
--- Số đơn mỗi khách (từ bảng Đơn hàng) ---
  ID_KhachHang  SoDonDaDat
0         KH01           2
1         KH02           2
2         KH03           1

--- Bảng tổng hợp chi tiêu và số đơn của khách hàng ---
  ID_KhachHang      TenKH  Tuoi  TongChiTieu  SoDonDaDat
0         KH01  Nguyễn An    25         5000         2.0
1         KH02  Trần Bình    40         8000         2.0
2         KH03   Lê Cường    17         1500         1.0
3         KH04  Phạm Định    33         6000         0.0
```