In [1]:

```
import pandas as pd
```

In [3]:

```
fp='C:/Users/ADMIN/Downloads/Day_9_banking_data.csv'
banking_data=pd.read_csv(fp)
```

In [5]:

```
# Display the first 5 rows of the dataset
print("First 5 rows of the dataset:")
print(banking_data.head())
```

```
First 5 rows of the dataset:
         Date       Account_Type    Branch Transaction_Type  \
0  2023-01-19      Fixed Deposit   Central     Loan Payment   
1  2023-01-16            Current    Uptown       Withdrawal   
2  2023-01-10            Current    Uptown     Loan Payment   
3  2023-01-18            Savings    Uptown     Loan Payment   
4  2023-01-14  Recurring Deposit  Suburban     Loan Payment   

   Transaction_Amount  Account_Balance
0              985.51          6839.59
1              641.43          8908.39
2             3363.85         12428.67
3             1914.60          5776.63
4             2788.57          4779.04
```

In [7]:

```
# Generate basic statistics of numerical columns
print("\nBasic statistics of numerical columns:")
print(banking_data.describe())
```

```
Basic statistics of numerical columns:
       Transaction_Amount  Account_Balance
count           20.000000        20.000000
mean          2705.829500      7967.766500
std           1429.829787      2770.248821
min            641.430000      2592.160000
25%           1482.085000      6460.127500
50%           2567.645000      7905.275000
75%           4121.525000      9127.702500
max           4683.640000     12836.510000
```

In [9]:

```
# Check for missing values
print("\nMissing values in the dataset:")
print(banking_data.isnull().sum())
```

```
Missing values in the dataset:
Date                  0
Account_Type          0
Branch                0
Transaction_Type      0
Transaction_Amount    0
Account_Balance       0
dtype: int64
```

In [11]:

```
import pandas as pd
```

In [13]:

```
fp='C:/Users/ADMIN/Downloads/Day_9_banking_data.csv'
banking_data=pd.read_csv(fp)
```

In [15]:

```
# Group by Account_Type and calculate required aggregations
account_group = banking_data.groupby('Account_Type').agg({
    'Transaction_Amount': 'sum',         # Total sum of Transaction_Amount
    'Account_Balance': 'mean'           # Average Account_Balance
}).rename(columns={
    'Transaction_Amount': 'Total_Transaction_Amount',
    'Account_Balance': 'Average_Account_Balance'
})
print("Aggregations by Account_Type:")
print(account_group)
```

```
Aggregations by Account_Type:
                   Total_Transaction_Amount  Average_Account_Balance
Account_Type                                                        
Current                            15052.57              9893.404000
Fixed Deposit                      14102.59              6120.380000
Recurring Deposit                  15179.99              7627.283333
Savings                             9781.44              9134.110000
```

In [17]:

```
# Group by Branch and calculate required aggregations
branch_group = banking_data.groupby('Branch').agg({
    'Transaction_Amount': ['count', 'mean']  # Count of transactions and average Transaction_Amount
}).rename(columns={
    'count': 'Total_Transactions',
    'mean': 'Average_Transaction_Amount'
})
# Flatten the multi-level column names
branch_group.columns = ['Total_Transactions', 'Average_Transaction_Amount']
print("\nAggregations by Branch:")
print(branch_group)
```

```
Aggregations by Branch:
          Total_Transactions  Average_Transaction_Amount
Branch
Central                    8                 2942.338750
Downtown                   3                 3188.703333
Suburban                   5                 2773.278000
Uptown                     4                 1786.345000
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js